

21 世纪通信网络技术丛书
——网络通信与工程应用系列

无线传感器网络技术与应用

陈林星 编著

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书介绍无线传感器网络技术,主要包括三个部分的内容:无线传感器网络概述,包括基本概念、发展历史、主要技术、网络设计主要影响因素;无线传感器网络技术,包括 MAC、路由、拥塞控制与可靠传输、数据融合、安全、定位、同步、中间件方面的技术;无线传感器网络应用与编程,包括应用设计原理、网络编程、分层编程技术、融合应用编程体系架构。

本书内容丰富、新颖,概念清楚,层次结构合理、明晰,涵盖了当前国际上无线传感器网络的主要研究成果及内容,可帮助读者尽快全面了解和掌握无线传感器网络技术。

本书可供从事无线传感器网络的科研人员、工程技术人员、院校师生,以及所有对此感兴趣的人士阅读和参考。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

无线传感器网络技术与应用 / 陈林星编著 .—北京:电子工业出版社, 2009.3

(21 世纪通信网络技术丛书——网络通信与工程应用系列)

ISBN 978-7-121-08409-6

I. 无… II. 陈… III. 无线电通信—传感器 IV. TP212

中国版本图书馆 CIP 数据核字(2009)第 030090 号

责任编辑:王春宁 特约编辑:刘 涛

印 刷:

装 订:

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1092 1/16 印张:25.75 字数:640 千字

印 次:2009 年 3 月第 1 次印刷

印 数:3 500 册 定价:69.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

出版说明

通信网络技术是当今发展最快、应用最广和最前沿的通信领域之一。通信技术发展到今天,已经不是传统意义上的充满神秘色彩的深奥技术了,它已经与日常的应用密不可分。可以说,网络的出现,使通信技术得以有了广阔的用武之地。正是由于有了固定电话网、移动通信网和 Internet,使通信技术的应用在这些平台上有了用武之地,渗透到了我们日常生活的方方面面。

为了促进和推动我国通信产业的发展,电子工业出版社通信分社特策划了一套《21 世纪通信网络技术丛书》。这套丛书根据不同的层面,又细分为三个系列:<移动通信前沿技术系列>、<3GPP LTE 无线通信新技术系列>和<网络通信与工程应用系列>。

<移动通信前沿技术系列>是从移动通信技术(3G 技术)的应用现状与发展情况出发,全面介绍当今移动通信领域涉及的关键技术与热点技术,例如,软件无线电;移动 IP 技术;移动数据通信;WCDMA;TD-SCDMA;cdma2000 移动通信系统网络规划与优化;智能天线技术;认知无线电技术;WiMAX, WiFi, ZigBee 宽带无线接入技术;UWB 技术;UMTS 技术;Ad Hoc 技术等。

<3GPP LTE 无线通信新技术系列>是以 3GPP 中 LTE 标准的关键技术在无线、宽带、高速、资源的有效管理和利用,以及在 B3G/4G 无线通信领域的应用为主。LTE 作为 3G 技术的一个重要的长期演进计划,代表了国际无线通信领域的最新发展需求和解决方案,例如:基于 OFDM 的上、下行(HSxPA)的多址接入技术、随机接入技术、多天线 MIMO 技术、多链路自适应技术、多播技术、功率控制技术、宽带无线网络的安全性、可移动性、可管理性;高效信源与信道编码和调制 MQAM 技术等。

<网络通信与工程应用系列>是以技术为先导,以构建网络的体系结构、标准、协议为目标所开展的对现代无线、移动、宽带通信网络的规划与优化,以及结合工程应用的方向所提出来的。例如,无线网状网、WLAN、无线传感器网络、B3G/4G 通信网工程设计与优化、卫星移动通信网、三网融合技术、网络新安全技术与策略、RFID 应用网络、下一代基于 SIP 的统一通信、光网络与光通信等。

本套丛书依托各高等院校在通信领域从事科研、教学、工程、管理的具有丰富的理论与实践经验的专家、教授;各科研院所的研究员;国内有一定规模和研发实力的科技公司的研发人员,以及国外知名研究实验室的专家、学者等组成编写和翻译队伍,力求实现内容的先进性、实用性和系统性;力求内容组织循序渐进、深入浅出、理论阐述概念清晰、层次分明、经典实例源于实践;力求很强的可读性和可操作性。

本套丛书的主要读者对象是广大从事通信网络技术工作的各科研院所和公司的广大工程技术人员;各高等院校的专业教师和研究生;刚走上工作岗位的大学毕业生;以及与此相关的其他学科的技术人员。

本套丛书从 2008 年上半年开始将陆续推出,希望广大读者能关注它,多对本套丛书提出宝贵意见与建议,欢迎通过电子邮箱 wchn@phei.com.cn 进行探讨、交流和指正,以便今后为广大读者奉献更多、更好的优秀通信技术类图书。

前 言

传感器技术、低功率电子学以及低功率 RF 设计技术的发展和进步使人们已经能够开发微型、能够通过无线网络相互连接、相对价廉的低功率传感器，这种传感器也叫做微型传感器（Microsensor）。无线传感器网络（WSN）技术能够革新很多场合下的信息采集和处理，代表提取环境数据、各种环境可靠监视（包括监视、机器故障诊断、化学/生物检测等）的一种新方式。大规模无线传感器网络由数千个、甚至数万个微型传感器组成，各个微型传感器分散在一个巨大场中，用来获取纹理细密的高精度感知数据。微型传感器通常依靠电池供电，相互之间进行无线通信。

无线传感器网络可以布置在恶劣、苛刻、复杂、甚至敌方的物理环境中（比如遥远地理区域或者有毒的城市地点、自然灾害区、战场敌方区、战场火力打击区等），也可以布置在人不易接近的环境中（比如大工厂、飞机内部、机器内部，甚至人体内部等）进行低成本地维护感知或者监视感知，更可以布置在人易于接近的环境中（比如人体表面各个部位、房间各个角落）进行各种状态监视。

无线传感器网络可以对大量感知信息（比如地震数据、声学数据、高分辨率图像等）进行分布式处理，提高感知数据的精确性。传感器组成网络后，能够累积感知数据，从而提供对环境的一个丰富的、多维的了解。此外，网络化传感器能够重点关注网络中其他传感器指出的关键事件（比如入侵者进入某个建筑物）。网络化传感器在面对各个传感器失效时也仍然能够继续发挥准确的作用。例如，假如网络中的一些传感器丢失某些关键信息，那么其他传感器就可以给这些传感器补充丢失的信息。

可以想象未来一组传感器节点构成 Ad Hoc 分布式处理网络，产生易于访问和高质量的有关真实世界的信息。每个传感器节点在网络中自动工作，不需要中央控制中心；每个传感器节点根据其承担的任务、当前拥有的信息以及所了解的计算资源、通信资源、能量资源、存储资源来做出决策。与孤立的各个传感器比较，网络化传感器有可能精确性更高，系统更加强壮和复杂。

无线传感器网络深入我们生活的每个环节、渗透社会的每个角落，有利帮助人类提高认识物理世界的深度、广度、精确性、及时性，加强和密切人类与物理世界的联系，大力提高人类对物理环境的远端监视和控制能力，所以无线传感器网络应用前景非常广阔。

无线传感器网络设计面临许多技术挑战，比如能量高效网络协议、网络拓扑控制、信号与信息的联合处理、任务分配、信息查询、安全、中间件、网络编程等，其中最重要的挑战是三个关键资源的高效综合利用：①能量，无线传感器节点主要依靠电池供电，电池不方便替换或者重复充电，且大多数情形下不可能替换或者重复充电，而且目前的电池供电能力也非常有限；②通信带宽，无线传感器网络的通信带宽相对于有线网络而言是非常有限的，一般只有几百千比特每秒；③计算能力，由于节能非常关键，所以一般情况下不会给传感器节点配置功能强大的微型处理器，而是采用低功耗、计算能力有限的微型处理器，因而不能运行复杂的网络协议。

无线传感器网络技术是一门跨学科的综合性的网络系统技术，除了涉及最基本的通信、计

计算机语言、编程知识和技术外，还涉及传感器网络所有应用领域的一些专门知识（比如医学、地震学、土壤学、农学等），这些特殊应用领域的专门知识对于设计高性能传感器网络应用是非常重要的。

本书系统地介绍了无线传感器网络技术，包括三个部分的内容：无线传感器网络概述、无线传感器网络技术、无线传感器网络应用与编程。其目的是为无线传感器网络设计者、研究人员、院校师生以及所有对此感兴趣的人士等全面、系统地理解和掌握无线传感器网络技术提供一些帮助。本书的编写安排如下：

第一部分“无线传感器网络概述”：包括 1 章（第 1 章）。介绍无线传感器网络的发展历史，阐述基本概念，分析无线传感器网络的主要技术及其网络设计影响因素，简单介绍了当前流行的传感器节点平台。

第二部分“无线传感器网络技术”：包括 11 章，其中 WSN 的 MAC 技术 3 章（第 2、3、4 章），WSN 路由技术 3 章（第 5、6、7 章），WSN 可靠传输技术 2 章（第 8、9 章），WSN 数据融合技术 1 章（第 10 章），WSN 安全技术 1 章（第 11 章），WSN 中间件技术 1 章（第 12 章）。各章安排如下：

第 2 章是无线传感器网络竞争类 MAC 协议，详细描述了三个典型的 WSN 竞争类 MAC 协议：传感器媒介访问控制协议（S-MAC）、超时 MAC 协议（T-MAC）、伯克利媒介访问控制协议（B-MAC）。

第 3 章是无线传感器网络分配类 MAC 协议，详细描述三个典型的 WSN 分配类协议：流量自适应媒介访问协议（TRAMA）、分布式随机时隙安排协议（DRAND）、功率高效与时延意识媒介访问协议（PEDAMACS）。

第 4 章是无线传感器网络混合类 MAC 协议，详细描述 WSN 时间同步技术和两个典型的 WSN 混合类 MAC 协议：斑马-MAC 协议（Z-MAC）、漏斗-MAC 协议。

第 5 章是无线传感器网络数据中心路由协议，详细描述了两个典型的 WSN 数据中心路由协议：协商式传感器信息分发协议（SPIN）、定向扩散。

第 6 章是无线传感器网络分层路由协议，详细描述了两个典型的 WSN 分层路由协议：低能量自适应分群分层协议（LEACH）、两层数据分发协议（TTDD）。

第 7 章是无线传感器网络地理位置路由协议，详细描述了 WSN 定位技术以及两个典型的 WSN 地理位置路由协议：贪婪地理路由算法、位置辅助泛洪协议（LAF）。

第 8 章是无线传感器网络端到端可靠传输协议，详细描述了 WSN 拥塞检测与预防技术（CODA）和两个典型的 WSN 端到端可靠传输协议：事件到中心节点的可靠传输协议（ESRT）、基于多电台虚拟中心节点的过载流量管理协议（SIPHON）。

第 9 章是无线传感器网络逐跳可靠传输协议，详细描述了 WSN 合成拥塞控制技术（FUSION）和两个典型的 WSN 逐跳可靠传输协议：慢分发快提取可靠传输协议（PSFQ）、下行数据可靠交付可扩展体系结构（GARUDA）。

第 10 章是无线传感器网络数据融合协议，详细描述了树状结构累积技术、不受应用约束的自适应数据累积技术（AIDA）、无结构累积技术（DAA+DW）与半结构累积技术（ToD）。

第 11 章是无线传感器网络安全，详细分析了 WSN 安全面临的障碍、WSN 安全要求，剖析 WSN 中的各种安全攻击，详细描述 SPINS 安全解决方案、LEAP+安全解决方案。

第 12 章是无线传感器网络中间件协议，分析了 WSN 中间件设计面临的挑战和困难，及其功能要求，详细介绍 ZebraNet 系统中的中间件系统（Impala）、无线传感器信息网络化体

系结构与应用中间件体系结构 (SINA)，其间介绍了 SINA 在车辆跟踪中的应用。

第三部分“无线传感器网络应用及编程”：包括 1 章（第 13 章）。概括了 WSN 在军事、环境、医疗卫生、家庭以及其他商业领域的应用；介绍了 WSN 应用设计原理；阐述了 WSN 网络编程问题，包括编程抽象、编程模型，比如 Kairos 编程模型、微型传感器网络虚拟机 (Mate)、采用属性状态机的无线传感器网络编程 (OSM)；详细描述了 WSN 分层编程技术、抽象任务图宏编程架构 (ATaG)。

本书的编写参阅了大量的研究文献和资料。在每章最后列出本章的主要参考文献。电子科技大学骆睿老师仔细审阅了本书第 2、3、4、7、12 章，并提出了许多改进之处。中国电子科技集团公司电子第 30 研究所高级工程师曾曦审阅了本书第 1、5、6、11、13 章；刘亮审阅了第 10 章；马蓉审阅了第 8 章；张虎审阅了第 9 章。此外作者还得到了谢青、刘英、陈曦、刘静、马先庆、刘伟、叶国宏、刘萍、李家国、王庆、王婷、曾令长、刘陶惠、罗永秀、曾晖、谢长富、周华等人的帮助。在本书的构思和写作过程中，以及本书的成功出版，作者一直得到了电子工业出版社、尤其是电子工业出版社通信分社王春宁博士的大力支持和帮助。作者在此一并表示由衷的感谢！

由于作者知识有限，本书难免会有缺陷，甚至错误。非常欢迎读者来文指出本书的缺点和错误。联系 E-mail: clx-clx-clx@163.com。

目 录

第 1 章 无线传感器网络概述	1
1.1 传感器网络的研究历史	1
1.1.1 早期的军用传感器网络研究	1
1.1.2 美军 DARPA 的分布式传感器网络研究计划	2
1.1.3 20 世纪 80 年代和 90 年代的军用传感器网络	3
1.1.4 21 世纪的传感器网络研究	4
1.2 WSN 基本概念	4
1.2.1 什么是 WSN	5
1.2.2 WSN 与 MANET 的异同	6
1.2.3 WSN 的通信体系结构	7
1.3 WSN 的主要技术	9
1.3.1 系统体系结构	9
1.3.2 网络与通信的控制	11
1.4 影响 WSN 设计的因素	18
1.4.1 容错	18
1.4.2 扩展性	19
1.4.3 价格	19
1.4.4 硬件限制	19
1.4.5 WSN 拓扑	20
1.4.6 WSN 工作环境	21
1.4.7 传输媒介	22
1.4.8 功耗	23
参考文献	25
第 2 章 无线传感器网络竞争类 MAC 协议	29
2.1 传感器媒介访问控制协议 (S-MAC)	29
2.1.1 能量浪费原因分析	29
2.1.2 S-MAC 协议概述	30
2.1.3 休眠的协调	32
2.1.4 避免旁听与消息分片传输	34
2.1.5 时延分析	36
2.1.6 S-MAC 协议实现	39
2.1.7 S-MAC 协议的性能	40
2.2 超时 MAC 协议 (T-MAC)	43
2.2.1 T-MAC 协议概述	43
2.2.2 T-MAC 基本协议	44

2.2.3	分群与同步	45
2.2.4	RTS 操作与 TA 选择	45
2.2.5	避免旁听	46
2.2.6	不对称通信	47
2.2.7	T-MAC 的性能	48
2.3	伯克利媒介访问控制协议 (B-MAC)	51
2.3.1	B-MAC 协议的设计与实现	51
2.3.2	寿命建模	53
2.3.3	参数	55
2.3.4	自适应控制	55
	参考文献	57
第 3 章	无线传感器网络分配类 MAC 协议	59
3.1	流量自适应媒介访问协议 (TRAMA)	59
3.1.1	TRAMA 协议概述	59
3.1.2	TRAMA 协议组成	60
3.1.3	访问方式与相邻节点协议	61
3.1.4	传输时间安排交换协议	62
3.1.5	自适应选举算法	64
3.1.6	TRAMA 的性能	66
3.2	分布式随机时隙安排协议 (DRAND)	69
3.2.1	TDMA 时隙分配问题定义	70
3.2.2	DRAND 算法详述	70
3.2.3	DRAND 正确性	72
3.2.4	DRAND 复杂性分析	73
3.2.5	DRAND 的性能	74
3.3	功率高效与时延意识媒介访问协议 (PEDAMACS)	79
3.3.1	PEDAMACS 协议概述	79
3.3.2	PEDAMACS 分组格式	80
3.3.3	本地拓扑建立阶段	80
3.3.4	AP 拓扑信息收集阶段	83
3.3.5	传输时间安排阶段	83
3.3.6	拓扑调整阶段	84
3.3.7	传输时间安排算法	84
	参考文献	88
第 4 章	无线传感器网络混合类 MAC 协议	91
4.1	斑马 MAC 协议 (Z-MAC)	91
4.1.1	时间同步协议 (TPSN)	92
4.1.2	Z-MAC 协议概述	94
4.1.3	相邻节点寻找与时隙分配	95
4.1.4	本地成帧	96

4.1.5	Z-MAC 协议的传输控制	97
4.1.6	发送规则	97
4.1.7	直接竞争通知	98
4.1.8	Z-MAC 传输时间安排的接收	100
4.1.9	本地时间同步	100
4.1.10	Z-MAC 协议的性能	101
4.1.11	Z-MAC 协议随机分析	103
4.2	漏斗-MAC 协议	105
4.2.1	漏斗问题	106
4.2.2	按需发送信标	107
4.2.3	面向中心节点的传输时间安排	109
4.2.4	定时与成帧	112
4.2.5	Meta-传输时间安排的广播	113
4.2.6	动态深度调整	113
4.2.7	漏斗-MAC 协议的测试床实验评估	116
	参考文献	120
第 5 章	无线传感器网络数据中心路由协议	122
5.1	协商式传感器信息分发协议 (SPIN)	122
5.1.1	SPIN 概述	123
5.1.2	Meta-Data	123
5.1.3	SPIN 消息	123
5.1.4	SPIN 资源管理	124
5.1.5	SPIN 实现	124
5.1.6	SPIN-1: 3 步握手协议	124
5.1.7	SPIN-2: 低能量门限的 SPIN-1	125
5.1.8	用于与 SPIN 比较的其他数据分发算法	126
5.1.9	SPIN 的性能评估	127
5.1.10	SPIN 小结	133
5.2	定向扩散	134
5.2.1	定向扩散的组成要素	134
5.2.2	命名	135
5.2.3	兴趣与梯度	135
5.2.4	数据传播	138
5.2.5	路径建立与路径裁剪的强化	139
5.2.6	定向扩散的分析评估	142
5.2.7	定向扩散的仿真评估	145
	参考文献	148
第 6 章	无线传感器网络分层路由协议	151
6.1	低能量自适应分群分层 (LEACH)	151
6.1.1	LEACH 协议体系结构	151

6.1.2	群首选择算法	152
6.1.3	分群算法	153
6.1.4	稳定状态阶段	154
6.1.5	LEACH-C: BS 建立分群	156
6.1.6	LEACH 的分析与仿真	156
6.2	两层数据分发协议 (TTDD)	160
6.2.1	两层数据分发	162
6.2.2	栅格结构	162
6.2.3	TTDD 转发	164
6.2.4	栅格维护	166
6.2.5	TTDD 开销分析	167
6.2.6	TTDD 的性能	170
6.2.7	TTDD 讨论	174
	参考文献	175
第 7 章	无线传感器网络地理位置路由协议	178
7.1	定位技术	178
7.1.1	距离测量与角度测量	178
7.1.2	位置计算	179
7.1.3	TPS 网络模型	179
7.1.4	TPS 定位方案	180
7.1.5	TPS 技术性能分析	183
7.2	贪婪地理路由算法	185
7.2.1	概述	186
7.2.2	基于 DT 的膨胀分析	188
7.2.3	贪婪转发 (GF)	190
7.2.4	有界 Voronoi 贪婪转发 (BVGF)	192
7.2.5	网络膨胀分析总结	196
7.2.6	基于概率通信模型的扩充	196
7.3	位置辅助泛洪协议 (LAF)	198
7.3.1	LAF 协议概述	198
7.3.2	采用 LAF 分发信息	201
7.3.3	LAF 中的资源管理	201
7.3.4	栅格维护开销	201
7.3.5	数据分发规程的完备性	202
7.3.6	LAF 节能分析	203
7.3.7	位置估计中的误差	204
7.3.8	LAF 的性能	204
	参考文献	206

第 8 章 无线传感器网络端到端可靠传输协议	210
8.1 事件到中心节点的可靠传输协议 (ESRT)	210
8.1.1 问题定义	210
8.1.2 评估环境	212
8.1.3 特性区域	214
8.1.4 ESRT 协议描述	215
8.1.5 拥塞检测	218
8.1.6 ESRT 协议对并发事件的处理	219
8.1.7 ESRT 协议的性能分析	222
8.1.8 ESRT 协议的仿真结果	223
8.1.9 ε 的正确选择	225
8.2 基于多电台虚拟中心节点的过载流量管理 (SIPHON)	225
8.2.1 拥塞检测与预防 (CODA)	226
8.2.2 虚拟中心节点寻找与可见度范围控制	232
8.2.3 SIPHON 拥塞检测	233
8.2.4 改变流量的传输路径	234
8.2.5 次网络中的拥塞	235
8.2.6 虚拟中心节点开销分析	235
参考文献	236
第 9 章 无线传感器网络逐跳可靠传输协议	239
9.1 合成拥塞控制技术 (FUSION)	239
9.1.1 拥塞崩溃的症状	239
9.1.2 逐跳流量控制	240
9.1.3 速率限制	241
9.1.4 MAC 层优先级化	241
9.1.5 应用自适应	242
9.2 慢分发、快提取可靠传输协议 (PSFQ)	242
9.2.1 PSFQ 协议概述	243
9.2.2 PSFQ 分发操作	245
9.2.3 PSFQ 提取操作	246
9.2.4 PSFQ 报告操作	248
9.2.5 单个分组消息的交付	249
9.2.6 PSFQ 的性能	249
9.3 下行数据可靠交付可扩展体系结构 (GARUDA)	252
9.3.1 面临的挑战	252
9.3.2 可靠性语义	253
9.3.3 GARUDA 的基本原理	254
9.3.4 单个分组或第一个分组的交付	257
9.3.5 即时构建 GARUDA 核	259
9.3.6 两阶段丢失恢复	260

9.3.7 其他可靠性语义的支持	261
9.3.8 GARUDA 的性能	263
参考文献	265
第 10 章 无线传感器网络数据融合技术	268
10.1 树状结构累积	268
10.1.1 分布式生成树算法	268
10.1.2 E-Span 树	269
10.2 不受应用约束的自适应数据累积 (AIDA)	270
10.2.1 AIDA 协议概述	271
10.2.2 AIDA 体系结构	271
10.2.3 AIDA 控制单元中的累积方案	272
10.2.4 AIDA 累积功能单元	275
10.2.5 AIDA 分组格式	275
10.2.6 AIDA 分组头开销分析	277
10.2.7 AIDA 节省分析	277
10.2.8 AIDA 的性能	278
10.3 无结构累积法与半结构累积法	281
10.3.1 数据意识任意组播 (DAA)	282
10.3.2 ToD 上的动态转发	286
10.3.3 性能分析	292
10.3.4 ToD 和 DAA 的性能	295
参考文献	298
第 11 章 无线传感器网络安全	300
11.1 WSN 安全概述	300
11.1.1 WSN 安全威胁模型	300
11.1.2 WSN 安全面临的障碍	300
11.1.3 WSN 安全要求	302
11.1.4 WSN 安全解决方案的评估	304
11.2 WSN 中的安全攻击	304
11.2.1 物理层安全攻击	305
11.2.2 链路层安全攻击	306
11.2.3 对 WSN 网络层 (路由) 的攻击	307
11.2.4 对传输层的攻击	310
11.3 SPINS 安全解决方案	310
11.3.1 符号	311
11.3.2 SNEP	311
11.3.3 μ TESLA	313
11.3.4 μ TESLA 详细描述	314
11.3.5 SPINS 实现	316
11.3.6 SPINS 性能评估	318

11.4	LEAP+安全解决方案	319
11.4.1	假设条件	319
11.4.2	LEAP+概述	319
11.4.3	单独密钥的建立	320
11.4.4	成对密钥的建立	321
11.4.5	分群密钥的建立	325
11.4.6	全网密钥的建立	325
11.4.7	本地广播认证	326
11.4.8	LEAP+安全分析	327
11.4.9	LEAP+性能评估	329
	参考文献	330
第 12 章	无线传感器网络中间件技术	334
12.1	WSN 中间件面临的挑战	334
12.2	WSN 中间件的功能要求	335
12.3	ZebraNet 系统中的中间件系统 (Impala)	335
12.3.1	ZebraNet 系统简介	336
12.3.2	ZebraNet 中间件体系结构	338
12.3.3	应用适配器	342
12.3.4	应用更新器	344
12.3.5	周期性操作调度	347
12.3.6	事件处理模型	348
12.3.7	Impala 网络接口	350
12.3.8	Impala 评估	353
12.4	传感器信息网络化体系结构 (SINA)	359
12.4.1	SINA 的功能组成	359
12.4.2	信息抽象	361
12.4.3	传感器查询与任务分配语言 (SCTL)	361
12.4.4	传感器执行环境 (SEE)	362
12.4.5	信息收集方法	362
12.4.6	应用举例	363
	参考文献	366
第 13 章	无线传感器网络应用及编程	368
13.1	传感器网络的应用	368
13.1.1	军事应用	368
13.1.2	环境应用	369
13.1.3	医疗卫生应用	370
13.1.4	家庭应用	371
13.1.5	其他商业应用	371
13.2	WSN 应用设计原理	373
13.2.1	设计方面	373

13.2.2 确定 WSN 操作坊式 376

13.3 WSN 网络编程..... 378

13.3.1 编程抽象 378

13.3.2 现有若干编程模型简介 379

13.4 分层编程与 ATaG 编程架构 381

13.4.1 WSN 的分层编程..... 381

13.4.2 抽象任务图编程架构（ATaG） 383

13.4.3 采用 ATaG 的应用开发方法 389

13.4.4 一个 ATaG 应用例子..... 390

参考文献..... 391

第 1 章 无线传感器网络概述

许多领域需要监视和测量各种物理现象[比如温度、液位、振动、损伤（张力）、湿度、酸度、泵、生产线的发电机、航空、建筑物维护等]，包括建筑工程、农林业、卫生、后勤、交通运输、军事应用等。有线传感器网络一直长期用于支持这种环境，直到最近也只是在有基础设施不可行的时候（比如偏僻区域、敌对环境）才使用无线传感器。有线传感器网络安装、停机、测试、维护、故障定位、升级的成本高，从而使得无线传感器网络（Wireless Sensor Network, WSN）很有吸引力。

最新技术发展已经使得人们能够生产智能、自治、能量高效并且可以大量使用的传感器，在地理区域中构成自组织和自愈 WSN。无线传感器技术成本大幅度下降，因而具有广泛的应用。随着 WSN 技术和其他相关技术的不断发展进步，WSN 将不断成熟，极有可能长期而显著地改变人类的日常生活。

WSN 技术是一门跨学科的新技术。下面首先介绍传感器网络的研究历史，然后介绍 WSN 的基本概念，比较 WSN 与 MANET 的异同，分析 WSN 网络设计的影响因素，简述 WSN 涉及的主要技术，最后详细描述 WSN 中的一个重要概念——覆盖范围。

1.1 传感器网络的研究历史

传感器网络的发展需要三个不同研究领域的技术：感知、通信和计算（包括硬件、软件、算法）。这三个领域的共同进步和其中某个领域的进步已经推动了传感器网络的研究和发展。初期的传感器网络包括空中交通控制使用的雷达网络。美国国家电网包含许多传感器，可以看做一种大型的传感器网络。这些系统具有专门的计算机和通信能力，并且在“传感器网络”术语流行之前就已经存在。

1.1.1 早期的军用传感器网络研究

就像很多技术那样，国防、军事应用是传感器网络研究和一个推动力。在冷战时期，美国就在战略区域布置了声学监视系统（Sound Surveillance System, SOSUS），用于检测和跟踪静默下的前苏联潜艇。SOSUS 是一种声学传感器（水下测声仪）系统，安装在海底。之后，美国开发了其他较复杂的声学网络，用于潜艇监视。现在美国国家海洋与大气管理局（National Oceanographic and Atmospheric Administration, NOAA）利用 SOSUS 来监视海洋事件，比如地震和海洋动物活动情况^[1]。在冷战时期，美国还开发和布置了防空雷达系统，用于保护美国大陆和加拿大。这种防空雷达系统已经经历了多年的发展，包括像传感器一样的航空器以及机载报警与控制系统（Airborne Warning and Control System, AWACS）飞机，现在还用于禁止毒品交易。

这些传感器网络一般采用分层处理结构，各层依次进行处理，直到有关感兴趣事件的信

息到达用户为止。在很多情况下，操作员在系统中起着关键作用。尽管早期军用传感器网络的研究重点是为了满足任务需求，比如声学信号处理和释义、跟踪、融合，但是早期军用传感器网络为现代传感器网络提供了一些关键处理技术。

1.1.2 美军DARPA的分布式传感器网络研究计划

传感器网络的现代研究起始于 1980 年左右美国国防高级研究计划署（Defense Advanced Research Projects Agency, DARPA）的分布式传感器网络（Distributed Sensor Networks, DSN）研究计划。此时，美国的 ARPANET（互联网的前身）也已经工作运行了好几年，具有 200 个综合性大学和研究机构的主机。TCP/IP 协议的发明者之一 R. Kahn 先生在互联网协议开发中发挥了关键作用，是 DARPA 信息处理技术办公室（Information Processing Techniques Office, IPTO）主任。R. Kahn 先生希望知道 ARPANET 通信技术是否能够延伸到传感器网络中。假定网络有很多分布在空间的低成本传感器节点组成，各个节点既相互协作，又各自独立工作，信息可以传递给任何需要的节点。

在当时的技术发展水平下，这是一个雄心勃勃的研究计划。这正好是个人计算机和工作站之前的事情，传感器的处理工作大都由小型机（比如运行 Unix 和 VMS 的 PDP-11 机、VAX 机）来完成，调制/解调器的速度为 300~9 600 baud/s，以太网也正好开始流行。

1978 年召开的分布式传感器网络专题研讨会^[2]确定了 DSN 的技术组成，包括传感器（声学）、通信（在资源共享网络公共应用上进行链路处理的高级协议）、处理技术和算法（包括传感器自定位算法）、分布式软件（动态可更改分布式系统和语言设计）。由于 DARPA 此时正在大力发起人工智能（Artificial Intelligence, AI）的研究，所以专题研讨会还包括提到了各种分布式问题解决技术以及运用 AI 来理解信号和访问态势。由于可用的现成技术很少，所以最后的 DSN 研究计划不得不解决分布式计算支持、信号处理、跟踪以及测试床方面的问题，选择分布式声学跟踪作为示范的目标问题。

卡内基梅隆大学（CMU）在 DSN 研究项目中的研究重点是提供网络操作系统，以便灵活、透明地访问容错 DSN 所需要的分布式资源。CMU 的研究人员开发了一个称做 Accent 的面向通信的操作系统^[8]，其原型支持透明网络化、系统重组以及重新装订。Accent 进一步发展成 Mach 操作系统^[9]，后者获得了相当大的商业成功。CMU 的其他研究成果包括支持活动通信计算的动态重新装订的网络进程间通信协议、构建分布式系统软件的接口技术规范语言、DSN 软件的动态载荷平衡和故障重组系统。在室内测试床上，通过与以太网连接在一起的信号源、声学传感器、VAX 计算机，演示了所有这些研究成果。

麻省理工学院（MIT）在 DSN 研究项目中的研究重点是基于已知信息的信号处理技术^[10]，以便采用信号提取和匹配技术、通过运用分布式麦克风阵列来跟踪直升机。信号提取技术认为信号由多电平信号组成，抑制较低电平（比如频谱）中的详细信息，提取较高电平（比如峰值）中的信息。MIT 的研究人员提供了一个概念性框架体系，用于研究信号处理系统，该系统类似于人们在交互式处理和解释真实世界信号时的系统。通过综合人工试探法，将这种方法用于缺乏模型的高信噪比率场合。此外，MIT 还开发了信号处理语言和交互式计算环境（Signal Processing Language and Interactive Computing Environment, SPLICE），用于 DSN 数据分析和算法开发；开发了推销主管助理，借助该助理和使用活动范围信息进行交互式基本频率评估。

推动分布式环境中的处理链、多目标跟踪技术发展比推动集中式跟踪技术发展要困难得多。假定测量与跟踪和估计关联，那么目标状态的测量与跟踪和估计的关联必须分散到各个传感器节点上。在 20 世纪 80 年代，加拿大的 ADS（Advanced Decision System）公司开发了一个多重假设跟踪算法，用于处理有关困难问题，包括目标密度高、目标丢失检测、伪告警等，以及剖析了分布式实现的算法。现在多重假设跟踪算法是困难跟踪问题的一个标准方法。

MIT 林肯实验室在 DSN 研究项目中开发了一个低速飞行航空器声学跟踪实时测试床^[12]，用于演示所开发的成果。传感器是声学阵列（9 个麦克风呈三个同中心三角形排列，最大的等边三角形的边长为 6 m）。采用 PDP11/34 计算机和一批处理器处理声学信号。中央计算机（用于目标跟踪）由三个 MC68000 处理器、256 KB 存储器、512 KB 共享存储器、一个用户操作系统组成。采用以太网和微波电台进行通信。图 1-1^[13]表示声学阵列（9 个麦克风）、移动汽车节点（其后背配有一个降音产生器）、设备机架（含汽车内的声学/跟踪节点和网关节点）。注意设备的体积，网络的几乎全部设备都是定制的。这就是 20 世纪 80 年代初期的水平。DSN 测试床与低速飞行航空器一起演示，演示中采用声学传感器和 TV 电视摄像机成功跟踪了低速飞行航空器。跟踪算法非常复杂，这是因为声音传播时延与航空器的飞行速度密切相关。

在 DSN 研究项目中开发的另一个 DSN 测试床是马萨诸塞州大学开发的分布式车辆监视测试床。这是一个根据经验研究分布式网络问题解决方法的研究工具。分布式、基于已知信息的问题解决方法采用功能精确的协作式体系结构，该结构由一个 Hearsay-II 节点（包含信息源的黑板结构）构成的网络组成。参考文献[14]介绍了其他的本地节点控制法。



图 1-1 1985 年前后 DSN 测试床的组成

1.1.3 20 世纪 80 年代和 90 年代的军用传感器网络

尽管早期的传感器网络研究人员已经研究了大量微型传感器问题，但是微型传感器技术仍然十分不成熟。军用系统规划人员很快认识到传感器网络的好处，传感器网络是网络中心战的一个关键组成部分^[15]。在平台中心战中，平台“拥有”专门的武器，专门武器又包含传感器，其结构非常严格、呆板。换言之，传感器和武器安装在一起，并且受到独立操作的孤立平台控制。在网络中心战中，传感器不一定安装在武器或者平台上，而是在通信网中相互协作，将信息传递给适当的“射手”。传感器网络通过多次观测、几何多样性与现象多样性、延伸的检测范围、较快的响应时间，能够提高检测性能和跟踪性能。此外，采用商用网络技术和公共网络接口降低了开发成本。

一个网络中心战的例子就是美国海军开发的联合作战能力系统（Cooperative Engagement Capability, CEC）^[16]。CEC 系统由多部收集空中目标数据的雷达组成。测量结果跟“具有报告职责”的处理节点有关，并且与处理所有兴趣测量结果的其他节点共享。由于所有节点实质上必须访问相同的信息，所以获得一个对统一军事作战必需的“公共作战图”。其他军用传感器网络包括反潜的声学传感器阵列，比如固定分布式系统（Fixed Distributed System, FDS）、高级可展开系统（Advanced Deployable System, ADS）；无人地面传感器系统（Unattended Ground Sensor, UGS）^[17]，比如远端战场传感器系统（Remote Battlefield Sensor System, REMBASS）、战术远端传感器系统（Tactical Remote Sensor System, TRSS）。

1.1.4 21 世纪的传感器网络研究

计算与通信的最新发展和进步已经极大地推动了传感器网络的研究，并且已经将其推动到比较接近原始预想的境界。基于微型机电系统（Micro-Electro-Mechanical System, MEMS）技术、无线网络技术以及廉价低功耗处理器的微型廉价传感器让人们能够布置无线 Ad Hoc 网络，开展各种各样的 WSN 应用。而且，DARPA 开始发起一个传感器网络研究计划（即 SensIT），用于支持最新技术进步。

DARPA 制定的 DARPA 传感器信息技术（Sensor Information Technology, SensIT）研究计划^[18]主要从事两个关键技术的研究和开发。第一个关键技术是开发新的网络技术。在战场环境中，传感器装置或者传感器节点应该随时可以按照 Ad Hoc 方式布置到高速动态环境中。在固定基础设施上开发的语音和数据网络技术不能满足战场环境下的运用。因此，SensIT 研究计划开发了适合于高速动态 Ad Hoc 环境下的新网络技术。第二个关键技术是网络化信息处理，即如何从展开的传感器网络中提取有价值、可信赖、及时的信息。这就意味着支持由这些传感器节点构建的分布式计算环境，以便在网内进行信号和信息处理、动态地查询传感器网络、动态地分配传感器网络任务。

SensIT 形成了跟当代传感器有关的新能力，诸如战术自动化安全系统（Tactical Automated Security System, TASS）^[19]之类的当代周长安全系统是专门可编程的。TASS 采用基于只发送节点和远距离检测的技术。SensIT 系统具有新能力有交互式、可编程、动态任务分配、动态查询等。SensIT 系统的多任务特性支持并行多用户。最后，由于传感器系统的检测距离短得多，所以软件和算法可以采用装置面临的威胁接近度来大力提高威胁检测和跟踪的精确性。软件和系统总体设计支持低时延、能量高效操作、内置自治能力和抗毁能力、操作检测概率低。因此，SensIT 节点网络系统能够支持网络内和网络外诸如高空侦察之类威胁的检测、识别、跟踪，以及支持目标确定和通信。

1.2 WSN基本概念

最近，微型机电系统（MEMS）、无线通信、数字电子学的发展和进步使得人们能够开发低成本、低功耗、多功能、体积小、短距离无线通信的传感器节点。这种微型传感器节点包括感知、数据处理、通信等组成部分，有力地支持基于大量节点共同协作的传感器网络思想。

1.2.1 什么是WSN

一个 WSN 由一组毫米般大小、设备齐全的 MEMS 装置组成，覆盖一个物理地理区域。这种微型装置包含传感器、无线发射机和接收机、电源，具有计算处理能力（即 CPU 能力）。例如，Smart Dust 研究计划的传感器节点（Mote 传感器）在一个只有几毫米的 MEMS 芯片上完成无线通信功能、传感器功能、电源、信息处理功能。

WSN 是传统传感器的重大进步，表现在以下几个方面：

① 传感器可以位于离真实现象较远的位置（即通过感知发觉的东西）。因此，要求大量传感器使用某种复杂技术区分环境噪声中的对象。

② 可以配置若干个只执行感知任务的传感器。需要仔细设计传感器的位置以及通信拓扑。传感器将所感知到的现象的时间序列发送给中心节点，中心节点对此进行计算以及数据融合。

一个 WSN 由大量密集布置在物理现象区域内或者附近的传感器节点组成。不必规划，也不必预先确定传感器节点的位置。一方面，可以将传感器节点随机布置在难以接近的地形区域或者赈灾作业环境中。但是，另一方面，这就意味着 WSN 协议和算法必须具有自组织能力。WSN 的另一个独特特点是传感器节点共同协作。传感器节点与内置处理器安装在一起，不是将原始数据发送给融合处理节点，而是利用其自身处理能力就地对原始数据进行简单计算，然后只将所要求的和部分处理过的数据发送给融合处理节点。

WSN 是智能的。有些 WSN 采用网内处理技术，将从源处收集的感知数据转变为比较抽象的累积高级数据，再将其发送给中心节点。将处理能力、存储、无线通信综合在一起意味着可以使用智能算法收集和分发数据。为许多应用规划大量传感器节点也意味着很大一部分 WSN 必须具有自组织能力。

典型的 WSN 直接与中心控制器或者卫星通信，传感器与控制器之间的通信是一跳通信。WSN 也可以由一组自治传感器节点或者终端组成，传感器节点构成一个多跳无线网络（即 Ad Hoc 网络），采用分布式方式维护连通性。当传感器节点之间的连通性由于节点移动性而随时变化时，这种 WSN 能够动态改变其拓扑。但是目前真正使用的大都是固定传感器节点。

直观上，基础设施越密集，WSN 就越有效，提供的数据精度就越高，用于累积的能量就越多。假如处理不当，那么密集 WSN 会引起传输碰撞和网络拥塞，从而导致时延增大、能量效率下降。WSN 的一个主要特征就是感知、通信、计算之间不存在明显的界限。互联网中的数据累积主要由端点来负责和完成，而在 WSN 中每个节点既是路由器又是数据源。

分布在物理环境中的传感器组成 WSN，包括作为可视传感器的摄像机、作为音频传感器的麦克风，以及能够感知超声波、红外线、温度、湿度、噪声、压力、震动的传感器。尽管单个传感器的感知范围有限，但是 WSN 通过综合许多传感器的感知数据能够覆盖较大地理空间，因此能够获得有关环境的不同而精确的信息。WSN 是一种正在涌现出来的计算平台，由大量低功率微型无线 Mote 传感器组成，每个 Mote 具有有限的计算、感知、通信能力。实现如下分布式 WSN 仍然是一个挑战：体积小、成本实在的传感器模块，高速、低时延、可靠网络基础设施，支持 WSN 简单而高效安装的软件平台，传感器信息处理技术。

由于诸如安全、隐密之类的社会问题，WSN 可能会成为一种破坏性技术，但是 WSN 各种新的、不同类型的应用是为了社会进步。防火中的环境监视，检测海平面腐蚀情况，研究

地震震动模式对桥梁、建筑物的影响。支持各种类型的监视，比如入侵者提前检测。无线传感器可以深嵌入到机器内，这是有线传感器不容易做到的，有线成本太高，灵活性有限，不能深嵌入到零件内，存在维护问题，不能移动。

WSN 编程困难，人工编程资源成本高。WSN 编程复杂性源于每个 WSN 节点能力（如处理能力、传输能力、存储能力）有限、能量资源、无线信道不可靠。因此，应用设计人员必须做出许多复杂的低级选择，设计支持路由、时间同步、节点定位、数据累积软件。但是，这种软件是按照复杂性、资源使用方法、通信模式，针对特定应用平衡而封装的，因而不能从一个应用直接移植到另一个应用中。不存在通用的 WSN 应用。WSN 的应用依赖性强于传统分布式应用。

1.2.2 WSN与MANET的异同

有两种主要类型的无线 Ad Hoc 网络：移动 Ad Hoc 网络（Mobile Ad Hoc Network，MANET）和 WSN。一个 MANET 就是一组通过带宽有限的无线链路连接在一起的自治移动路由器及其有关主机。每个节点被看做一个个人信息装置，比如个人数字助理（Personal Digital Assistant，PDA），配备有相当复杂的无线收发信机。这种节点是全移动的，MANET 无线网络拓扑可能迅速变化且不可预测。MANET 既可独立工作，也可以连接到较大的互联网中。许多因素，比如不断变化的链路质量、传播路径损耗、衰落、多用户干扰、功耗、拓扑变化等，都会引起 MANET 网络协议设计复杂性明显提高。另外还需要仔细考虑 MANET 的安全、时延、可靠性、人为干扰、故障恢复等问题。

一个 WSN 由许多分散在某个地理区域内的传感器节点组成。每个传感器节点具有无线通信能力、智能化程度足够高的数据处理和网络处理能力。WSN 可以布置在遥远的地理位置上，要求最低的建立成本和管理成本。将 WSN 与较大网络（比如互联网）或者无线基础设施网络综合在一起，能够拓宽覆盖范围，以及开发 Ad Hoc 网络的潜在应用领域。采用多跳通信将感知信息中继给中心节点。中心节点是传感器节点，具有连接外部网络（比如互联网）的网关功能，通常通过中心节点来分发感知信息，如图 1-2 所示。

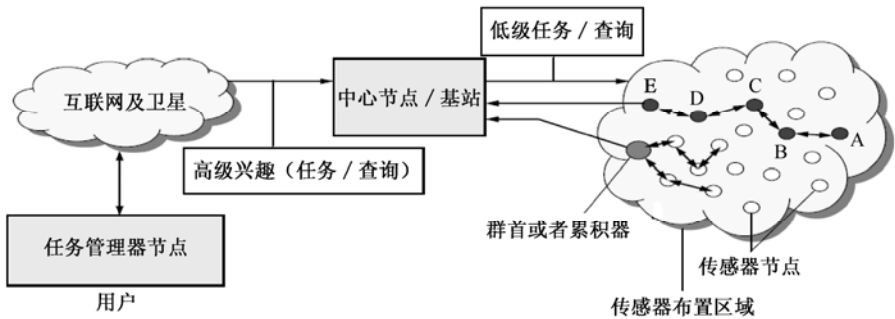


图 1-2 WSN 概况

WSN 在许多基本方面不同于 MANET。将 WSN 看成一个大规模多跳 Ad Hoc 网络对于许多实际应用可能是适当的。配置网络到工作状态的通信开销极高。WSN 网络节点数比 MANET 网络节点数多达数个量级，传感器节点密集布置、易于失效。传感器节点主要采用广播通信，而大多数 MANET 采用对等（Peer-to-Peer，P2P）通信。在 WSN 中极少进行端到

端节点间的信息交换。WSN 节点的能量、计算能力、存储器容量非常有限，可能没有全网统一的 ID。WSN 应用范围广，包括环境监视、敏感装置、远程数据收集和分析。在 MANET 和 WSN 中，节点既是主机又是路由器，按照自组织和自适应方式工作。无基础设施无线网络领域的研究和发展一直在快速推进，还需要在这个研究方向进行更加努力，实现大规模广泛应用。当前的传感器硬件资源和能量是有限的，但是硬件技术将会迅速发展，成本将会快速降低，因此 WSN 最终可能具有 MANET 的特性。

由于大量传感器节点密集布置在一起，所以相邻节点之间距离可能非常近。因此，WSN 多跳通信的功耗低于传统单跳通信的功耗。传感器节点可以保持低发送功率，例如秘密军事行动需要非常低的功率。多跳通信也能够克服远距离无线通信所遇到的一些信号传播效应问题。

传感器节点的最重要限制之一是低功耗要求。传感器节点的功耗源有限、并且通常不能替换。因此，尽管传统网络的目标是提供高服务质量（Quality of Service, QoS），但是 WSN 协议的重点必须主要放在节能问题上，必须内嵌折中机制，让端用户选择以较低吞吐量或者较高传输时延为代价，从而延长网络寿命。

WSN 区别于 MANET 的一个重要方面是传感器处理收集的数据。在 WSN 中，最终目标是检测/估计一些感兴趣的事件，而不是通信。融合单个传感器或者多个传感器的数据有利于事件处理和事件检测性能提高。数据融合要求发送数据和控制消息，是 WSN 体系结构的组成部分之一。传感器数据联合处理是 WSN 区别于 MANET 又一个因素。累积多个传感器的数据有利于提高感兴趣事件的检测速率，这又要求发送数据和控制消息，对网络体系结构提出某种程度的约束。

WSN 的各种应用需要无线移动 Ad Hoc 网络技术。尽管已经提出了许多无线 Ad Hoc 网络的协议和算法，但是这些协议和算法不适合 WSN 的独特特性和应用要求。

概括起来，WSN 和 MANET 之间的区别如下：

- ① 一个 WSN 的传感器节点数量比一个 MANET 的节点数量高出数个量级。
- ② 传感器节点密集布置。
- ③ 传感器节点易于失效。
- ④ WSN 网络拓扑变化非常频繁。
- ⑤ 传感器节点主要采用广播通信方式，而大多数 MANET 以点对点通信方式为基础。
- ⑥ 传感器节点的功率、计算能力、存储容量非常有限。
- ⑦ 传感器节点开销高、数量大，因此传感器节点可能没有全球识别码（ID）。

1.2.3 WSN的通信体系结构

一个传感器节点由图 1-3（a）所示的四个基本部分组成：感知单元、处理单元、收发信机和功率单元。感知单元通常由两个子单元组成：传感器（数量、种类可变）和模/数转换器（ADC）。ADC 将传感器根据其观测到的物理现象而生成的模拟信号转换成数字信号，然后将其送给处理单元。处理单元通常连着一个存储单元，管理各个传感器节点共同完成所承担的感知任务而必需的相互协作规程。收发信机将传感器节点连接到网络上。传感器节点最重要的组成单元是功率单元，功率单元受到诸如太阳能电池之类的功率提取单元的支持。传感器节点还可能包括其他子单元（比如定位系统、功率发生器、移动管理器），许多 WSN 应用都依赖这些子单元。大多数 WSN 路由技术和感知任务需要高精度的位置信息。因此，传感

器节点常常包含定位系统。在要求传感器节点完成所分配任务时，有时可能需要移动管理器来移动传感器节点。

传感器节点通常分散在图 1-2 所示的传感器场中，每个传感器节点均能够收集数据，并将其发送给中心节点和端用户。收集到的数据通过无基础设施的多跳体系结构传递给中心节点，然后再到达端用户，如图 1-2 所示。中心节点能够通过互联网或者卫星与任务管理器节点进行通信。

图 1-3 (b) 给出了中心节点和所有传感器节点使用的协议栈。协议栈将功率意识和路由意识组合在一起，将数据与网络协议综合在一起，在无线传输媒介上进行能量高效通信，支持各个传感器节点相互协作。协议栈由应用层、传输层、网络层、数据链路层、物理层、功率管理平面、移动管理平面、任务管理平面组成。根据感知任务，可以在应用层上建立和使用不同类型的应用软件。传输层帮助维护 WSN 应用所需要的数据流。网络层解决传输层提供的数据的传输路由问题。由于环境噪声以及传感器节点可能是移动节点，所以 MAC 协议必须具有能量意识能力，能够使与相邻节点广播的碰撞达到最低程度。物理层解决简单而又强壮的调制、发送、接收技术问题。此外，功率管理平面、移动管理平面、任务管理平面分别监视传感器节点之间的功率、移动、任务分配，帮助传感器节点协调感知任务和降低总功耗。

功率管理平面管理每个传感器节点如何运用其能量。例如，传感器节点接收到其中一个相邻节点的一条消息后，可以关闭接收机，这样可以避免接收重复的消息。一个传感器节点剩余能量较低时，可以向其相邻节点广播，通知它们自己剩余能量较低，不能参与路由功能，而将剩余能量用于感知任务。移动管理平面用于检测和记录传感器节点的移动状况，因而总是维护返回到用户的路由，传感器节点能够连续不停地跟踪其相邻传感器节点。传感器节点获知其相邻传感器节点后，就能够平衡其能量和任务处理。任务管理平面平衡和安排特定区域内的感知任务。并不要求区域内的全部传感器节点同时执行感知任务。因此，有些传感器节点根据其能量等级而执行比其他传感器节点较多的感知任务。功率管理平面、移动管理平面、任务管理平面是必需的，这样各个传感器节点才能够一起高效地工作，在移动 WSN 中传输数据，共享资源。如果没有功率管理平面、移动管理平面、任务管理平面，那么每个传感器节点只能单独工作。从整个 WSN 来看，若传感器节点能够相互协作，则网络效率更高，因而 WSN 的寿命更长。下面针对 WINS^[38]、Mote^[31]、μAMPS^[43]中的协议栈进一步比较讨论图 1-3 (b) 所示的协议栈。

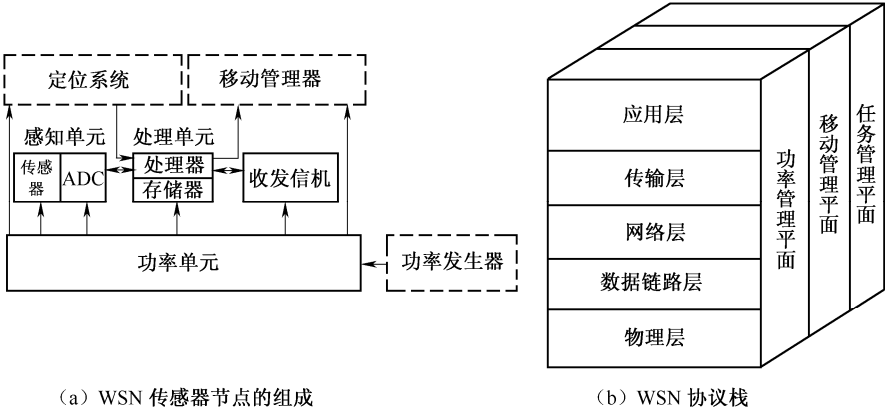


图 1-3 WSN 协议栈和节点组成

无线综合网络传感器（Wireless Integrated Network Sensor, WINS）将微型传感器技术和低功率信号处理、计算、低成本无线网络技术综合到一个紧凑的系统中，提供分布式网络和对传感器节点、控制、处理器的互联网访问。由于传感器节点数量庞大，所以 WINS 网络利用传感器节点之间的短距离传输提供多跳通信服务，以及使能耗最低化。WINS 将数据传输给用户的方法与图 1-2 所示的体系结构相同。传感器节点（即 WINS 节点）检测环境数据，然后通过 WINS 节点逐跳发送感知数据，直至感知数据传递到达中心节点（即 WINS 网关）。所以，根据图 1-2 所示的体系结构，WINS 节点就是传感器节点 A、B、C、D、E。WINS 网关通过传统网络服务（比如互联网）与用户通信。WINS 网络协议栈由应用层、网络层、MAC 层、物理层组成。WSN 需要一整套低功率协议来解决其各种约束问题。

灵敏尘粒（Mote）就是传感器节点，体积小、重量轻，所以可以安装在目标上甚至空中浮动物体上。灵敏 Mote 传感器节点采用 MEMS 技术进行视距通信和感知，在白天可以采用太阳能电池积聚能量，采用视距传输与基站收发信机或者其他灵敏 Mote 传感器节点进行通信。通过比较灵敏 Mote 传感器节点通信体系结构和图 1-2 所示的体系结构，灵敏 Mote 传感器节点通常直接与基站收发信机（即中心节点）通信。也可采用对等通信，但是由于存在“隐含节点”问题，所以可能发生媒介访问碰撞问题。在灵敏 Mote 传感器节点中综合的各个协议层是应用层、MAC 层、物理层。

设计 WSN 协议和算法的另外一种方法受到物理层需求的驱动^[43]。应该根据选定的物理层组件（比如微处理器类型、接收机类型）开发 WSN 的协议和算法。 μ AMPS 无线传感器节点自低层到上层的设计法（简称为由下往上法）针对应用层、网络层、MAC 层以及物理层的重要性，将其紧凑地与传感器节点硬件综合在一起，见图 1-3（b）。 μ AMPS 无线传感器节点也按照图 1-2 所示的体系结构与用户进行通信。参考文献[43]比较了不同的方案，比如时分多址（TDMA）与频分多址（FDMA）的对比，二进制调制与多进制调制的对比等。由下往上法指出：WSN 算法必须具有硬件意识，能够运用微处理器和收发信机的特性使传感器节点功耗达到最低。这有助于在设计不同类型传感器节点时沿着用户自定义解决方案方向发展。开发出不同类型传感器节点必然导致得到不同类型的 WSN。这就有可能导致开发出不同类型的协作算法。

1.3 WSN的主要技术

WSN 的发展依赖许多技术，比如硬件、系统软件、网络通信。WSN 的难题之一是一个应用对 WSN 的要求不同于另一个应用对 WSN 的要求。要求 WSN 易于安装、自我识别、自行诊断、可靠、时间意识、位置意识。对传感器节点和网络的精确性和采样速率，很难定义一个通用要求。图 1-4 给出了 WSN 的主要技术。下面简单分析支持 WSN 的各种技术（硬件技术除外），重点介绍系统和网络通信技术。

1.3.1 系统体系结构

WSN 系统体系结构包括操作系统、目标定位与跟踪、定位、时间同步、安全、编程等内容。

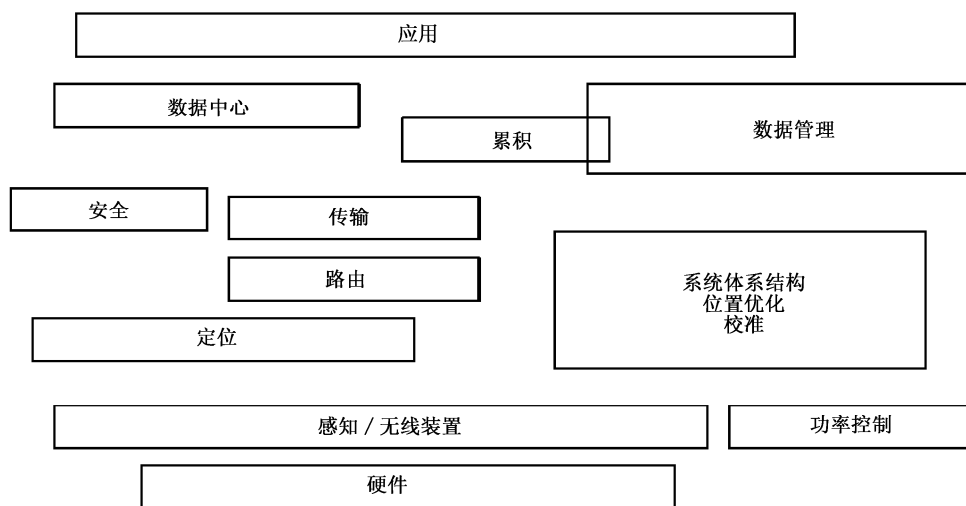


图 1-4 WSN 技术

1. 系统软件与平台

系统软件与平台包括操作系统和编程语言方面的技术。要求操作系统支持并行操作，不同数据流必须同时传递。操作系统必须提供高效模块化，必须将硬件和特定应用组件综合在一起，处理开销和存储开销要低。用户能够从操作系统提供的组件集中选择应用需要的最小组件。各个组件能够并行执行，同时等待事件最少、资源消耗最低。本章稍后简单介绍当前最流行的 TinyOS 操作系统和一些较典型的传感器节点硬件平台。

2. 定位

当获得某个传感器节点的感知信息时，其物理位置就是获取该感知信息的地点。因此，传感器节点的位置是分析传感器感知信息的必要信息，此外还可以用于许多 WSN 协议和算法（比如路由协议）。

精确定位是许多 WSN 应用的关键技术，包括监视网络、机器传感器、基于位置的 WSN 路由协议、利用移动传感器的太空与环境智能监视。第 7 章将介绍 WSN 定位技术。

3. 时空相关

传感器节点密集布置、连续观测物理现象是 WSN 的特点之一。由于 WSN 网络拓扑节点密度高以及物理现象的特性，所以传感器的观测结果在空间和时间上高度相关。

- ① 空间相关：典型的 WSN 应用为了可靠覆盖而要求空间密集布置传感器。
 - ② 时间相关：事件跟踪可能要求传感器节点周期性观测和发送感知数据。
- 时空相关与 WSN 联合协作性对高效通信协议极为有利。

4. 节点位置优化

覆盖范围就是一个传感器节点的空间感知范围。传感器节点之间必须相互协调，降低冗

余度，必须考虑感知任务的通信距离和其他特征。本章稍后将进一步深入讨论 WSN 覆盖范围问题。

5. 时间同步

时间管理、WSN 有关时间的操作是时戳事件、控制与操作周期、网络同步操作、传输信道访问与控制等所必需的。第 4 章将详细介绍 WSN 时间同步协议 TPSN。

6. 安全

因为 WSN 可能布置在敌方环境中工作，所以安全应该是 WSN 设计过程中不可忽略的一项内容，而不是在 WSN 设计之外来考虑。要求网络技术提供低时延、抗毁能力强、安全的网络。传感器可以用于敌人后方，所以要求 WSN 低检测概率通信。同样的原因，WSN 应该具有抗入侵和哄骗能力。

安全是 WSN 重要技术之一。为了保护 WSN 的安全，需要分析 WSN 面临各种攻击时存在的脆弱性，分析解决 WSN 安全问题时面临的 WSN 固有挑战，也需要分析针对 WSN 可能采取的攻击手法、攻击策略等，最后需要高效、节能、快速、准确的安全解决方案。第 11 章将介绍 WSN 安全技术。

1.3.2 网络与通信的控制

WSN 网络通信与控制方面的技术包括通信方法、节能、拥塞控制、拓扑管理、路由、建模等。由于以下原因，互联网和无线 Ad Hoc 网络的体系结构不能用于 WSN：

- WSN 传感器节点比 Ad Hoc 网络多得多；
- 传感器节点失效率很高；
- 传感器节点的能力（计算、通信、存储）有限；
- 传感器节点的能量有限（主要依靠电池供电，电池一般不能或者不易替换）；
- 极少使用应答分组。

因此，WSN 的体系结构必须：

- 同时具有能量意识和路由意识；
- 将数据与网络协议综合为一体；
- 能量高效通信；
- 与相邻节点共享任务。

1. 节能通信协议

传感器节点布置在物理环境中必须保持足够长的工作寿命，因此节能非常重要。节能机制直接影响网络寿命。一般地，网络寿命定义为网络能够执行感知功能和能够将数据发送给中心节点的时间周期。处在网络寿命期间，有些传感器节点可能变成无用节点（比如物理受损、缺乏能量的节点），或者可能布置新的传感器节点。经常采用的一个机制就是安排传感器节点的网络活动时间，以便使多余的传感器节点尽可能多地进入休眠方式，以及尽可能长时间地处在休眠方式。为了设计这样的机制，必须解决以下问题：①每个传感器节点依据什么

规则决定是否进入休眠方式？②传感器节点何时做出这样的决定？③传感器节点应该在休眠方式下保留多长时间？

另外一种降低能耗的方法就是使感知距离最小，同时又能满足感知覆盖范围目标。假如允许自适应感知距离，那么处在活动方式下的传感器应该动态地将其感知距离调整到一个最小值，以便满足总的感知目标。传感器的通信距离也可以采用同样的原理。假如传感器能够调整其通信距离，那么使通信距离最短同时又能维持网络连通性要求，就能够达到节能的目的。

传感器节点有体积、重量、成本的限制，这些限制将影响资源的可用性。传感器节点电池能量有限，处理能力和通信能力也均有限。由于在很多应用中不能替换电池，所以低能耗是需要考虑的一个关键因素，不仅在硬件设计和体系结构设计中需要考虑能耗，而且在网络体系结构各个层次上的算法设计和网络协议设计中都需要考虑能耗。包括从物理层、信道编码、媒介访问控制层到路由、传输、应用层的优化和综合考虑。因此，网络寿命最大化是一个重要的网络设计目标。运用最少传感器节点（特别是在确定性传感器节点布置方法中）也是一个明确的网络设计目标。

传感器节点的电台有四种工作状态：发送、接收、空闲、休眠，传感器节点可以处在这四种工作状态中的任何一种。空闲状态就是收发信机既不发送、也不接收的那段时间，休眠状态指电台被关闭的那段时间。根据参考文献[55]的介绍，通过对 WINS Rockwell 地震传感器功率用法的分析而指出：发送状态的功耗在 $0.38\sim 0.7\text{ W}$ ，接收状态的功耗为 0.36 W ，空闲状态的功耗为 0.34 W ，休眠状态的功耗为 0.03 W ，感知任务的功耗为 0.02 W 。一个有趣的观测结果是：接收方式和空闲方式需要的能量可能与发送方式相同，而在传统的无线 Ad Hoc 网络中，发送能量可能是接收能量的 2 倍。另外一个观测结果是通信/计算所需能量比，该比值可能高达 1 000 以上（比如对于 Rockwell WINS，该比值为 $1\,500\sim 2\,700$ ）。因此，需要采取本地数据处理、数据融合、数据压缩。通过传输时间安排机制正确选择每个传感器节点的电台工作状态，从而建立起既降低网络能耗同时又达到减少执行覆盖范围任务的活动传感器的数量的一个重要方法。有时，传输时间安排机制的目标也是维护活动传感器之间的连接。

MAC 层对通信协议能量效率，特别是对采用低占空因数电台的 WSN 的能量效率起着最重要的作用。根据 MAC 协议的工作机理，WSN 的 MAC 协议分成竞争类、分配类、混合类（竞争与分配的混合）三大类。第 2 章将详细描述传感器媒介访问控制协议（S-MAC）、超时空 MAC 协议（T-MAC）、伯克利媒介访问控制协议（B-MAC）三个典型的竞争类 MAC 协议。第 3 章将详细描述流量自适应媒介访问协议（TRAMA）、分布式随机时隙安排协议（DRAND）、功率高效与时延意识媒介访问协议（PEDAMACS）三个典型的分配类 MAC 协议。第 4 章将详细描述斑马-MAC 协议（Z-MAC）、漏斗 MAC-协议两个典型的混合类 MAC 协议。

2. 拥塞控制

有线网络通常采用端到端机制和网络层机制进行拥塞控制。但是，由于在不同无线链路上同时进行的发送相互影响、相互干扰，无线信道质量随着时间的推进而变化大，所以有线网络的拥塞控制方法不能解决无线网络的拥塞问题。WSN 以广播作为主要通信机制。

主要有两个原因引起 WSN 拥塞。第一个原因是分组到达速率大于分组服务速率。这在接近中心节点的传感器节点上是很可能发生的，因为越接近中心节点的传感器节点承载的组

合上行流量越大。第二个原因是链路级性能，比如竞争、干扰、比特误码率。第二个原因造成的拥塞发生在无线链路上。

WSN 拥塞直接影响能量效率和应用 QoS。例如，拥塞可能引起缓存器溢出、导致链路利用率下降，而缓存器溢出又可能导致排队时延增大、分组丢失率上升。分组丢失不仅会引起可靠性下降、应用 QoS 变差，而且还浪费有限的节点能量。假如使用竞争类 MAC 协议（比如 CSMA）共享无线资源，那么链路拥塞会引起发送碰撞，而发送碰撞既浪费能量又导致分组服务时间增大。因此，必须有效控制 WSN 拥塞，力求避免拥塞或者减轻拥塞。

在无线环境中，拥塞和比特误码都会造成分组丢失，分组丢失会引起端到端可靠性和 QoS 下降，从而进一步降低能量效率。造成分组丢失的其他原因包括节点失效、错误路由信息、过时路由信息、能量耗尽等。提高源节点发送速率或者采取基于重传的丢失恢复技术可以解决分组丢失问题。ESRT 采用提高源节点发送速率，对于没有分组可靠性要求的事件驱动应用能够很好保证事件可靠性。但是与丢失恢复比较，提高源节点发送速率不是能量高效法。丢失恢复法更加主动、能量效率更高，可以在链路层和传输层同时实现。链路层丢失恢复逐跳进行，传输层丢失恢复通常按照端到端方式进行。丢失恢复法由丢失检测、丢失通知、重传恢复三个机制组成。

为了设计高效传输协议，必须考虑几个因素，包括网络拓扑、应用的多样性、流量特点、资源限制。WSN 的两个最主要限制是分布在不同地理位置上的传感器节点间的能量和公平性。传输协议必须提供高能量效率和灵活的可靠性，假如有必要的话，则还要提供吞吐量、分组丢失率、端到端时延方面的 QoS。

由于拥塞控制和丢失恢复机制直接影响能量效率、可靠性、应用 QoS，所以 WSN 传输协议应该具有拥塞控制和丢失恢复机制。通常有两种方法实现 WSN 传输协议。第一种方法是独立设计拥塞控制、丢失恢复的协议或算法。现有的大多数 WSN 传输协议采用这种方法分别独立解决拥塞控制问题、丢失恢复问题。采用这种方法且一般使用模块设计，要求可靠性的应用可以只调用丢失恢复算法，要求拥塞控制的应用只调用拥塞控制算法。例如，CODA 是拥塞控制协议，PSFQ 提供可靠传输。联合使用 CODA 和 PSFQ 能够提供 WSN 传输协议所要求的全部功能。第二种方法是设计和实现一个完整传输协议，以集成方式同时提供拥塞控制和丢失控制。例如，STCP 在一个协议中同时实现拥塞控制和灵活的可靠性。对于不同的应用，STCP 提供不同的控制策略，既保证了应用要求，又提高了能量效率。

第一种方法将一个问题分解成多个子问题，易于灵活处理，第二种方法优化拥塞控制和丢失恢复，因为 WSN 中的丢失恢复和拥塞控制常常是相关的。例如，竞争型无线链路易于引起分组丢失，若是联合使用 CODA 和 PSFQ，则能够同时提供拥塞控制和可靠性。

第 8 章详细描述 WSN 端到端可靠传输技术，包括事件到中心节点的可靠传输协议 (ESRT)、基于多电台虚拟中心节点的过载流量管理协议 (SIPHON) 两个典型的 WSN 端到端可靠传输协议，以及 WSN 拥塞检测与预防技术 (CODA)。第 9 章详细描述 WSN 逐跳可靠传输技术，包括慢分快发提取可靠传输协议 (PSFQ)、下行数据可靠交付可扩展体系结构 (GARUDA) 两个典型的 WSN 逐跳可靠传输协议，以及 WSN 合成拥塞控制技术 (FUSION)。

3. 网络控制与拓扑管理

影响 WSN 拓扑管理的因素很多，包括应用、网络操作方式、网络性能（例如能耗、能量效率、寿命等）、移动性、无线链路质量、节点失效、增加新传感器节点等。

WSN 中的传感器节点为了正确操作,必须具备网络信息。每个传感器节点必须知道其相邻传感器节点的身份识别码和位置,才能够支持处理和相互协作。对于事先规划好的网络,网络拓扑常常是事先知道的。对于 Ad Hoc 网络,必须实时建立网络拓扑,以及随着传感器节点失效或者布置新传感器节点而周期性地更新网络拓扑。对于移动网络,由于网络拓扑在不断变化,所以需要提提供某种机制,用于传感器节点(固定的、移动的)之间的相互寻找。因为每个传感器节点只与其相邻传感器节点交互,所以通常不需要全网信息。

WSN 网络必须具有网络资源(如能量、带宽、处理能力)的处理能力,这些网络资源是动态变化的,系统应该自动工作,按需改变其配置。由于 Ad Hoc 网络没有事先规划好的连接,所以必须根据算法和软件按需建立连接。因为通信链路不可靠、阴影衰落可能导致链路中断,所以软件和系统设计应该产生所要求的可靠性。这就要求研究提供足够冗余所必需的诸如网络规模、链路数量、节点数量。相对于自由空间,地面网络的无线传输质量随着通信距离增大而下降得更快,这就意味着通信距离和能量必须进行最佳管理。协议设计必须内部化,不需要操作员干预。

4. 路由

通过布置足够数量的传感器节点,提供路径冗余,以及采用寻找合适路径的算法,确保抗毁性和环境自适应。扩散路由法依靠相邻节点的信息,可用于解决这个问题,但是这种方法可能不能达到空间分布式无线网络的信息理论容量^[47]。另外一个设计问题是研究诸如网络规模、每平方英里节点密度之类的系统参数对时延、可靠性、能量之间平衡的影响。

WSN 网络层的主要目标是寻找能量高效路由建立方法和将传感器节点的数据可靠中继到中心节点的方法,使 WSN 寿命达到最长。

WSN 的路由问题极富挑战性,主要是因为 WSN 具有不同于当代通信网络和无线 Ad Hoc 网络的几个独特特点:①不可能为使用大量传感器节点的 WSN 建立全网寻址方案。因此,基于 IP 的经典协议不能应用到 WSN 中;②相对于典型通信网络,WSN 的几乎所有应用都要求感知数据流从多个区域(数据源)传递到一个特定的中心节点;③因为多个传感器节点位于所观测现象的附近,因而可能产生相同的数据,所以 WSN 中产生的数据流冗余度高,路由协议必须利用 WSN 感知数据流高冗余度,提高能量和带宽的利用率;④传感器节点的发射功率、自身能量、处理能力、存储容量非常有限,因此需要仔细管理网络资源。

由于上述差异,已经提出了许多新的路由算法来解决 WSN 中数据传递的路由问题。这些路由机制不仅考虑了 WSN 的上述特点,而且还考虑了应用要求和体系结构要求。其中大部分路由协议尽管有些差异,但是根据网络流量和 QoS 意识,大致分成三种类型:①数据中心路由[第 5 章,包括协商式传感器信息分发协议(SPIN)、定向扩散两个数据中心路由协议];②分层路由[第 6 章,包括低能量自适应分群分层协议(LEACH)、两层数据分发协议(TTDD)两个分层路由协议];③位置路由[第 7 章,包括贪婪地理路由算法、位置辅助泛洪协议(LAF)两个地理位置路由协议]。数据中心路由协议是基于查询的路由协议,中心节点给预定区域发送查询消息,然后等待该区域内传感器节点发送来的数据。数据中心路由方法依赖所需数据的命名,对数据进行命名有助于排除许多冗余传输。分层路由协议的目标是对传感器节点进行分群,群首可以进行数据累积,减少数据量,节省能量,同时提供可扩展性。位置路由协议利用传感器节点的位置信息将数据中继到所需的区域,而不是中继到整个网络,其中包括基于通用网络流模型的路由协议,寻求满足一定 QoS 要求

及路由功能的路由协议。

WSN 路由协议的性能与体系结构密切相关：

① 网络动态性。一个 WSN 由传感器节点、中心节点、被监视事件三个主要部分组成。大多数 WSN 体系结构都假定传感器节点是静止的。另一方面，在有些情况下必须支持中心节点或者群首（网关）的移动。由于路由稳定性变成一个重要优化因素，以及能量、带宽的原因，所以给移动节点发送路由消息以及移动节点接收路由消息更富挑战性。根据应用，感知事件可以是动态的，也可以是静态的。例如，在目标检测/跟踪应用中，事件（现象）是动态的；而预防火灾的森林监视应用则是静态事件。网络用于监视静态事件时可以按照反应式方式工作，只是在报告事件时产生事件流。大多数应用中的动态事件要求周期性报告事件，结果产生较多的事件流需要传输给中心节点。

② 节点布置。另一个需要考虑的是节点的拓扑展开问题。这是个由应用决定的问题，并且影响路由协议的性能。节点展开或者是确定性的或者是自组织的。若是采用确定性展开，则通过人工布置传感器节点，数据通过预先确定的路径传递。但是采用自组织展开，则传感器节点随机扩散在场中，按照 Ad Hoc 方式建立基础设施。在 Ad Hoc 基础设施中，中心节点或者群首的位置对于能量效率和性能非常关键。当传感器节点分布不均匀时，最佳分群变成能量高效网络操作的一个迫切问题。

③ 能量考虑。在创建基础设施期间，路由建立过程受到能量考虑的极大影响。存在障碍物时，无线电台的发射功率与距离的平方或者距离的更高次幂成正比，所以多跳路由的能耗比直接通信低。但是，多跳路由存在一定的拓扑管理开销和媒介访问控制开销。假如所有传感器节点都非常接近中心节点，那么可以采用直接路由。在大多数情况下，传感器节点随机扩散在兴趣场中，多跳路由是必需的。

④ 数据交付模型。根据 WSN 的应用，到达中心节点的数据交付模型可以分成连续、事件驱动、查询驱动、混合四种类型。在连续数据交付模型中，每个传感器节点周期性发送数据。在事件驱动和查询驱动数据交付模型中，在发生一个事件或者中心节点产生一个查询的时候触发数据的发送。有些网络运用混合数据交付模型，将连续、事件驱动、查询驱动模型综合在一起。路由协议受到数据交付模型的极大影响，特别是能耗最低化对路由问题影响更加严重。例如，在栖息地监视应用中连续将数据发送给中心节点，因此分层路由协议是最有效的路由协议。这是因为这种应用产生冗余度甚高的数据，可以在传输给中心节点的路由上进行累积，既减少流量又节省能量。

⑤ 节点能力。在 WSN 中，各种功能可能与传感器节点有关。一个 WSN 的所有传感器节点可以是同类传感器节点，具有相同的计算能力、通信能力以及功率。但是，根据应用，某个传感器节点可以专门承担某个特定功能任务（比如中继、感知、累积）。一个传感器节点同时负责中继、感知、累积三个任务可能会很快地耗尽其能量。

WSN 也可以包含不同类型传感器，从而引起有关数据传输路由的多种技术问题。例如，有些应用可能需要运用不同类型传感器来监测周围环境的温度、压力、湿度，采用声学信号检测移动目标的运动，捕捉移动目标的图像或者视频轨迹。这些特殊传感器要么独立配置，要么按需使用的普通传感器中包含这些特殊传感器的功能。这些特殊传感器按照不同速率产生数据，受到不同服务质量要求的约束，遵循多种（混合）数据交付模型。因此，异种环境使得数据传输路由更富挑战性。

⑥ 数据累积/融合。因为传感器节点可能产生冗余度甚高的数据，所以可以对多个传感

器节点产生的相似分组进行累积，减少在无线媒介上发送的分组数量。数据累积就是采用抑制（排除相同的分组）、最小、最大，以及平均之类的累积函数运算组合来自不同数据源的数据。每个传感器节点可以完成其中一些或者全部累积函数，允许传感器节点管理网内数据整理和缩减。计算的能耗低于通信的能耗，通过数据累积可以节省大量能量。在许多路由协议中已经采用数据累积技术来实现能量效率和流量优化。在有些 WSN 体系结构中，所有数据累积功能全部由能力更强的专门传感器节点来完成。采用信号处理技术也可以实现数据累积，此时数据累积称为数据融合，传感器节点通过降低噪声以及采用一些信号组合技术能够产生更加精确的信号。

数据累积与网络体系结构密切相关。在平面网络中，每个传感器节点起着相同作用、配备有近似相同的电池能量，数据累积通常由数据中心路由来完成。中心节点承担较重的通信和计算任务，因此其能量消耗较快。中心节点失效则导致 WSN 功能性崩溃。在分层网络中，分层数据累积涉及在一些特殊节点进行数据融合，这些节点减少发送给中心节点的消息量，从而提高网络能量效率。在分群网络中，传感器节点将其数据发送给自己的群首，群首累积其群内所有传感器节点的数据，将累积得到的精炼数据发送给中心节点。群首可以通过远距离传输或者群首间多跳传输与中心节点进行通信，改善网络能量效率。在树状 WSN 中，将传感器节点组织成一棵树，树的中间节点执行数据累积，将累积数据的精简表示发送给树根节点。这种数据累积适用于涉及网内数据累积的应用。例如，核电厂内的核辐射监视应用，最大核辐射强度提供最为有用的该核厂安全信息。表 1-1 对分层网络与平面网络中的数据累积进行了概括性比较。

表 1-1 分层网络与平面网络中的数据累积对比

分 层 网 络	平 面 网 络
群首或者树叶节点执行数据累积	多跳路径上的各个节点执行数据累积
在整个网络中建立分群的开销	只在有数据需要发送的区域建立数据累积路由
即使一个群首失效，网络仍然能够工作	中心节点失效可能导致整个网路崩溃
传感器节点将其数据发送给自己的（因此相距较近）群首节点，因此时延较小	传感器数据通过多跳路径传递给中心节点，因此时延较大
路由结构简单、但不是最佳路由	最佳路由有保证，但是存在额外开销
可以使用异类传感器节点，群首节点分配高能量	不能利用异类传感器节点来提高能量效率

数据累积按照能量高效、最低数据时延方式致力于从传感器收集最重要数据。数据时延对于很多应用（比如环境监测）都是非常重要的，数据的新鲜程度是时延敏感应用的一个重要因素。开发能量高效数据累积算法、提高网络寿命非常关键。决定 WSN 能量效率的因素包括网络体系结构、数据累积机制、路由协议、MAC 协议等。

① 能量效率：WSN 的功能性应该尽可能地延长。对于一个理想数据累积方案，每个传感器节点在每轮数据采集中应该消耗相同的能量。一个数据累积方案若是能够使 WSN 的功能性延长到最大程度，则是能量高效的。假定所有传感器节点同等重要，则应该使每个传感器节点的能耗最低。网络寿命是对 WSN 能量效率的定量描述。

② 网络寿命：网络寿命定义为在网络中失效传感器节点所占比例达到 α （由系统设计员确定）之前完成的数据累积循环次数。能量效率和网络寿命对于提高能量效率达到延长网络寿命是同义的。例如，假如某种应用要求所有传感器节点一起工作的时间非常关键，那么网

络寿命定义为在第一个传感器节点耗尽其能量之前完成的数据累积循环次数。其主要思想就是按照能量均匀消耗方式进行数据累积。

③ 数据精度：数据精度定义与 WSN 具体应用有关。例如，对于目标定位应用，中心节点的目标位置估计决定数据精度。

④ 时延：时延定义涉及有关数据传输、路由、数据累积的时延，按照数据在中心节点的接收时刻与该数据在源节点的产生时刻之差来测量。网络寿命、数据精度、时延是评估数据累积算法的最重要性能指标。

第 10 章将详细描述 WSN 数据融合技术，包括树状结构累积技术、不受应用约束的自适应数据累积技术（AIDA）、无结构累积技术（DAA+DW）与半结构累积技术（ToD）。

5. 信号与信息的联合处理

WSN 网络中的各个节点相互协作、共同收集和处理数据，产生有用信息。信号与信息的网络联合处理是一个新的研究领域，跟分布式信息融合有关。重要技术问题包括节点之间的信息共享程度、节点如何融合其他节点的信息。处理来自多个传感器的数据通常会提高性能，但是需要较多的通信资源（因此包括能量）。类似地，在低层（比如原始信号）传输信息时，很少丢失信息，但是需要较多带宽。因此，在使用微型传感器时，在信号与信息的联合处理中，需要考虑性能和资源利用之间的许多平衡问题。

传感器节点接收到另一个节点的信息后，必须将该信息与本地信息融合在一起。融合方法包括从挑选最佳结果的简单规则到基于模型的技术，基于模型的技术考虑了信息如何产生的问题。另外，性能和强壮性之间也需要综合平衡。简单融合规则强壮，但不能达到最佳性能，而较复杂、较高性能的融合规则可能对低层模型比较敏感。在网络化环境中，信息可能经过多条路径而到达一个节点。融合算法应该识别被融合的冗余信息，避免同一个信息被多次融合。在由大量功能强大传感器节点组成的网络中，采用数据来源连续跟踪法。但是数据来源连续跟踪法可能不适用于处理能力和通信资源均有限的 WSN 网络。

WSN 经常用于目标检测、跟踪、识别。当一个小区内出现多个目标的时候，数据关联是一个非常重要的问题。每个节点必须将其自己的环境测量结果与单独每个目标关联。此外，一个节点检测到的目标必须与其他节点检测到的目标关联，以便避免重复和进行融合。最佳数据关联计算成本很高，通信带宽需求很高。因此，分布式数据关联就是性能和资源利用之间的一种综合平衡，要求分布式数据关联算法适应 WSN。

其他处理问题包括如何满足任务时延和可靠性要求以及如何使 WSN 工作寿命达到最大。廉价传感器节点构成的密集网络可能允许空间采样，不需要高开销算法。这些算法必须是异步的，这是因为处理器速度和通信能力可能不同，甚至消失和重新出现。传感器节点不断确定的数据结果必须精确度越来越高，当精确度达到足够高时，就结束数据确定过程。

6. 任务分配与查询

传感器场就像数据库，具有很多独特特性。数据是从环境中动态获取的，而不是操作员输入的；数据分布在各个传感器节点中，地理上分散的各个传感器节点通过不可靠链路连接在一起。这些特性使得传感器场这个数据库更富挑战性，特别是对于战场环境要求低时延、实时、高可靠性的军事应用更是如此。

用户有一个简单接口来交互式地给 WSN 分配任务和查询 WSN 是非常重要的。例如，入

网接口是可以接收语音输入的手持单元。用户应该能够控制信息的访问，比如，操作优先权和目标类型，同时隐藏单个传感器节点的细节。面临的一个挑战是开发查询和任务分配语言以及可以快速查找的数据库。其他挑战包括开发高效分布式查询与任务编辑和布置机制、数据组织机制、数据存储机制。

移动平台能够承载传感器节点和查询装置。因此，在没有基础设施的条件下，移动装置和固定装置之间的无缝互连互通互操作是 WSN 的一个关键而独特要求。例如，一个航空查询装置发出一个查询，然后告诉地面 WSN 自己 1 min 后将飞越某个特定位置，查询的响应应该安全地、悄悄地传递通过敌占区。

7. 建模

需要通过建模提前估计 WSN 性能以及分析 WSN 在真实环境中操作的技术，这有助于研究 WSN 在真实环境中的理论性能。例如，在理想设置下，已经证明严格地址中心路由协议能够大幅度地提高性能。

8. 组管理

协作组是一个很有用的概念，可以根据地理邻近关系、网络角色、资源有效性进行分组。但是采用分布式方式进行协作组管理需要可靠通信和所有传感器节点的一致同意。

1.4 影响WSN设计的因素

WSN 设计受到许多因素的影响，包括容错、扩展性、产品成本、工作环境、WSN 拓扑、硬件限制、传输媒介、功耗等。很多研究人员针对这些因素进行了深入的研究，下面对这些研究成果进行总结。在实际工作中需要全面、综合、系统地考虑驱动 WSN 和传感器节点设计的各种因素。这些因素非常重要，因为它们指导 WSN 协议和算法的设计。此外，可以使用这些影响因素比较不同的设计方案。

1.4.1 容错

有些传感器节点由于缺乏能量而可能失效或者中断正常功能，也可能受到物理损伤或者环境干扰。传感器节点失效不应该影响 WSN 承担的总体任务。这就是可靠性问题或者容错问题。容错是在传感器节点失效条件下继续维持 WSN 功能而不被中断的能力^[42, 53]。运用泊松分布将一个传感器节点的可靠性 $R_k(t)$ 或者容错模拟为时间间隔 $(0, t)$ 内功能正常的概率：

$$R_k(t) = \exp(-\lambda_k t)$$

式中， λ_k 表示传感器节点 k 的失效率， t 表示时间周期。

将协议和算法设计成满足 WSN 要求的容错能力。假如传感器节点所处环境干扰弱，那么就可以将网络协议放宽松一些。例如，将传感器节点布置在一间房子内，用于连续监视湿度和温度，由于这种环境下的 WSN 不容易被损坏或者受到环境干扰，因此容错要求较低。又如，假如将传感器节点布置在战场上，用于战场监视和检测，此时传感器的感知数据很重要，并且传感器节点易受敌方破坏，因此容错要求必须很高。所以容错能力取决于 WSN 的

应用, 必须根据这一要求设计和开发 WSN。

1.4.2 扩展性

在观测场中布置数百个, 甚至数千上万个传感器节点用于观测、捕捉以及研究某种物理现象。根据 WSN 的应用, 极端情况下可能需要数百万个传感器节点。新的 WSN 方案必须能够适应这么庞大的传感器节点而正常、高效地工作, 必须利用 WSN 的高密度特性。节点密度从一个区域内少数几个传感器节点到数千个传感器节点, 区域的直径可以小于 $10\text{ m}^{[26]}$ 。可以按照如下公式计算节点密度^[23]

$$\mu(R) = (N\pi R^2)/A$$

式中, N 表示区域 A 内布置的传感器节点数量, R 表示无线传输距离, $\mu(R)$ 主要表示区域 A 内每个传感器节点的传输覆盖范围内的传感器节点数量。

此外, 可以使用一个区域内的传感器节点数量来表示节点密度。节点密度取决于 WSN 应用所要求布置的传感器节点数量。对于机器诊断应用, 节点密度是一个 $5\text{ m} \times 5\text{ m}$ 区域内大约 300 个传感器节点; 对于车辆跟踪应用, 节点密度是每个区域内大约 10 个传感器节点^[43]。一般情况下, 节点密度可以是每立方米 20 个传感器节点。对于 WSN 家庭应用, 一个家庭可能包含 24 个家用装置, 其中每个家用装置均包含传感器节点^[36]; 假如将传感器节点嵌入到家具和其他各种各样家设中, 那么所需传感器节点数量还要增大。对于动植物栖息地监视应用, 每个区域 25~100 个传感器节点^[24]。一个人通常包含数百个传感器节点, 传感器节点可以嵌入到眼镜、衣服、鞋、手表、珠宝、人体内, 当人坐在露天大型运动场观看篮球、足球、棒球等比赛时, 节点密度就非常高。

1.4.3 价格

因为 WSN 由大量传感器节点组成, 所以单个传感器节点的成本对于整个网络的合理成本是非常重要的。假如 WSN 的成本比布置传统 WSN (一般是有线 WSN) 高出许多, 那么 WSN 就不是成本合理的系统。因此, 每个传感器节点必须保持低成本。达到最新技术发展水平的技术是一个蓝牙无线系统的成本低于 10 美元^[40]。一个 PicoNode 的目标成本低于 1 美元^[39]。为了实现切实可行的 WSN, 一个无线传感器节点的成本也应该低于 1 美元。蓝牙电台是低成本装置, 但是一部蓝牙电台的成本比一个传感器节点的目标价位高出 10 倍。传感器节点还有其他组成部分, 比如感知单元、处理单元等。此外, 根据 WSN 的应用, 传感器节点可能配置定位系统、移动管理器、功率发生器。因此, 在给定功能、并要求成本比 1 美元低许多的条件下, 传感器节点的成本是一个极富挑战性的问题。

1.4.4 硬件限制

一个传感器节点通常由感知单元、处理单元、收发信机、功率单元四个基本部分组成, 如图 1-3 (a) 所示。要求所有功能单元模块集成在一个火柴盒般大小的模块中。为了使传感器节点能够飘浮在空中, 甚至要求传感器节点的体积小于 1 cm^3 ^[38]。除了体积要求, 传感器节点还有一些其他严格要求。传感器节点必须^[31]: ①能耗极低; ②在高容量密度条件下工作; ③产品成本很低, 可管理; ④自治、无人照看也可工作; ⑤具有环境自适应能力。

由于传感器节点常常是人难以接近的,所以 WSN 的寿命依赖传感器节点能量资源的寿命。由于传感器节点体积的限制,能量也是一种稀缺的宝贵资源。例如,灵敏 Mote 传感器微粒的总储能为 $1\text{ J}^{[38]}$ 。对于 WINS^[45],平均系统总电流必须小于 $30\text{ }\mu\text{A}$,才能够提供足够长的工作寿命。WINS 节点通常依靠锂电池(直径为 2.5 cm 、厚 1 cm)供电。采用能量提取技术有可能延长 WSN 的工作寿命,能量提取技术从环境提取能量^[40]。比如,太阳能电池就是能量提取技术的一个例子。

传感器节点的收发信机组成单元可以是像灵敏 Mote 传感器微粒那样的被动或者主动的光学装置,也可以是无线射频装置。无线通信需要调制、带通、滤波、解调以及复接电路,从而使传感器节点更复杂和成本更高。此外由于传感器节点的天线比较接近地面,因此两个传感器节点之间的发送信号的路径损耗与两者之间距离的四次方成正比。但是,正在进行的大多数 WSN 研究项目首选无线通信,这是因为 WSN 传输的分组较少,数据速率较低(通常低于 1 Hz)^[40],通信距离短而频率复用高。这些特性也使得 WSN 有可能使用低占空因数无线收发信机装置。但是,能量效率和低占空因数电路设计目前仍然富有挑战性。目前商用电台开电、关电均要消耗不少的能量^[43],因此目前商用电台技术(比如蓝牙)的效能不能满足 WSN 的要求。

尽管处理器正在变得越来越小而其计算能力却越来越强,但是传感器节点的处理单元和存储器单元仍然是稀缺资源。例如,灵敏 Mote 传感器微粒原型^[35]的处理单元是 4 MHz Atmel AVR8535 微控制器、 8 KB 指令闪存、 512 B 的 RAM 以及 512 B 的 E^2PROM ,而采用的 TinyOS 操作系统需占用 $3\text{ }500\text{ B}$ 操作系统代码存储空间,因此用户可用程序代码存储空间只有 $4\text{ }500\text{ B}$ 。另一款传感器节点原型(即 μAMPS 无线传感器节点)的处理单元采用 $59\sim 206\text{ MHz}$ SA-1110 微控制器。 μAMPS 无线传感器节点运行多线程 $\mu\text{-OS}$ 操作系统。

大多数感知任务需要位置信息。传感器节点通常都是随机布置和无人照看的,所以必须与定位系统一起共同工作。许多 WSN 路由协议也需要定位系统。常常假定每个传感器节点配备有一个全球定位系统(Global Positioning System, GPS),定位精度低于 5 m 。研究指出^[41]:给所有传感器节点配备 GPS 对于 WSN 是不可行的,而是给若干个经过专门选定的传感器节点配备 GPS,同时帮助其他传感器节点寻找其地理位置。

1.4.5 WSN拓扑

那些人不能接近、无人照看的传感器节点易于频繁失效,造成 WSN 拓扑维护是一个具有挑战性的任务。在整个传感器场中布置数百个甚至数千个传感器节点。传感器节点之间相距几十英尺。节点密度可能高达 $20\text{ 个节点}/\text{cm}^{2[43]}$ 。传感器节点高密度展开要求仔细处理网络拓扑维护问题。下面分析三个阶段的拓扑变化和拓扑维护问题。

1. 展开前和展开阶段

可以采用投掷法或者逐个布置法将传感器节点布置到传感器场中。比如,可以采用如下方式布置传感器节点:

- 采用飞机定点、定区投掷;
- 采用大炮、火箭,甚至导弹布置;

- 采用弹射器（船甲板等）投掷；
- 布置在工厂内；
- 人工或者机器逐个布置。

尽管通过仔细制定传感器节点布置工程计划可以预防许多传感器节点出现人不能接近、无人照看的问题，但是在布置传感器节点的起始阶段仍然必须：

- 降低安装成本；
- 排除任何形式的事先组织和预先计划；
- 提高安排的机动性和灵活性；
- 提高自组织能力和容错能力。

2. 展开后阶段

在 WSN 布置完毕之后，由于传感器节点发生如下变化而引起网络拓扑变化：

- 传感器节点的位置；
- 传感器节点的可达性（由于干扰、噪声、移动障碍物等）；
- 传感器节点的可用能量；
- 传感器节点故障；
- 传感器节点承担的作业任务。

传感器节点可以固定不动。但是，由于能量耗尽或者节点被毁，所以定期或者经常发生传感器节点失效的问题。WSN 可能包含高速移动的传感器节点。此外，传感器节点和 WSN 承担动态任务，可能是人为故意干扰的目标。因此，WSN 拓扑在展开之后易于频繁变化。

3. 额外节点重新布置阶段

可以在任何时候重新布置新的传感器节点，替换发生了故障的传感器节点；或者由于任务的动态变化而重新布置额外的传感器节点。增加新的传感器节点需要重新组织 WSN。对于包含大量节点、并且具有极严格能耗限制的 Ad Hoc 网络，处理其频繁的网络拓扑变化需要特定的路由协议和拓扑管理与控制协议。

1.4.6 WSN工作环境

将传感器节点密集布置在所观测物理现象的附近或者范围内。因此，传感器节点经常处在偏僻而遥远的地理区域内独立工作，无人照看。传感器节点可以在

- 繁忙的十字路口工作；
- 在大型机器内部工作；
- 在海底工作；
- 在扭花装置内部工作；
- 在刮龙卷风时在海面工作；
- 在受到生化污染的场地中工作；
- 在敌方防线外的战场内工作；
- 在住宅或者大型建筑物内部工作；

- 在大型货仓内工作；
- 绑缚到动物身上工作；
- 固定到高速行驶车辆、火车、舰船上工作；
- 在排水沟或者河流中工作；
- 在人体身上或者体内工作；
- 在空中平台（飞机、航空航天装置等）上工作；
- 嵌入到家具内部工作；
- 飘浮在空中工作。

这些 WSN 工作环境同时也说明了传感器节点将在什么条件下工作。传感器节点能够在高压力的海底工作，在诸如残骸、战场之类的苛刻环境中工作，在诸如飞行器发动机喷嘴之类的极热环境、诸如北极之类的极冷环境中工作，在诸如故意干扰之类的噪声极强的环境中工作。

1.4.7 传输媒介

在多跳 WSN 中，各个通信节点通过无线媒介相互连接。这些链路可以由无线电波、红外线、光构成。为了全球操作 WSN，所选择的传输媒介必须是全世界可用媒介。

无线连接的一种选择是使用工业科学卫生（Industrial Scientific Medical，ISM）频带。大多数国家使用 ISM 频带进行通信时不需要许可证。国际频率分配表指定了可以用于 ISM 应用的一些频带，其中包括电台条例 S5 款（卷 1），见表 1-2。

表 1-2 ISM 应用频带

频 带	中 心 频 点
6 765~6 795 kHz	6 780 kHz
13 553~13 567 kHz	13 560 kHz
26 957~27 283 kHz	27 120 kHz
40.66~40.70 MHz	40.68 MHz
433.05~434.79 MHz	433.92 MHz
902~928 MHz	915 MHz
2 400~2 500 MHz	2 450 MHz
5 725~5 875 MHz	5 800 MHz
24~24.25 GHz	24.125 GHz
61~61.5 GHz	61.25 GHz
122~123 GHz	122.5 GHz
244~246 GHz	245 GHz

其中有些频带已经用于无绳电话通信系统和无线局域网（Wireless Local Area Network，WLAN）。对于 WSN，需要体积小、低成本、超低功率收发信机。参考文献[37]研究指出，一定的硬件约束条件、天线效率和功耗之间的折中平衡限制选择超高频频率（UHF）作为传感器收发信机的载波频率，欧洲建议采用 433 MHz 的 ISM 频带和北美建议采用 915 MHz 的 ISM 频带。参考文献[36]和[50]研究了这两个 ISM 频带的传感器收发信机设计问题。采用 ISM 频带的主要优点是电台频率不需要许可证、可分配频谱宽、全世界通用，不必局限于某个特

定标准,因此在 WSN 中实现各种节能策略较为方便。另一方面,存在各种各样的规定和限制,比如功率限制、来自现有应用的有害干扰。因此,ISM 频带也称为非受控频率。

当前大多数传感器节点的硬件都是以 RF 电路设计为基础。 μ AMPS 无线传感器节点采用兼容蓝牙的 2.4 GHz 收发信机和综合频率合成器。参考文献[46]介绍的低功率传感器装置采用工作于 916 MHz 的单信道 RF 收发信机。为 WINS 框架体系也采用无线链进行通信。

WSN 节点之间的另一种可能通信方式是采用红外线。红外线通信不需要许可证,抗电气装置干扰能力强。基于红外线的收发信机价格便宜,易于研制。现在的许多掌上电脑、PDA 以及移动电话都提供红外线数据通信接口。红外线通信的主要缺点是要求发送方和接收方之间处于视距范围。因此,红外线不宜作为 WSN 的传输媒介。

灵敏 Mote 传感器微粒是自动感知、计算、通信系统,采用光媒介进行通信传输,采用两种传输方案:采用三面直角棱镜后向反射器(Corner-Cube Retroreflector, CCR)的被动传输,采用激光管和可控镜的主动通信。对于前一种方案,Mote 传感器不需要内置光源,采用三面镜子结构(CCR)发送数字高或者数字低。后一种方案采用内置激光管和主动可控激光通信系统给预定接收机发送严格准直射偏振光束。

WSN 不同寻常的应用要求使得其传输媒介选择更富有挑战性。例如,潜艇应用可能要求使用水中传输媒介。为此,可以采用能够穿透水面的长波。敌方地面应用或者战场应用可能遇到易于发生误码的信道和较强的干扰。此外,传感器节点天线不高,辐射功率较低。因此,传输媒介选择必须结合高效而强壮的编码与调制方案,对信道特性的巨大差异进行高效模拟。

1.4.8 功耗

无线传感器节点是微型电子装置,只能装配有限的能源(<0.5 Ah, 1.2 V)。在有些应用场合,不能进行能源补给。因此,传感器节点的寿命非常依赖于电池的寿命。在多跳 Ad Hoc WSN 中,每个节点起着数据源和数据路由器的双重作用。少数几个节点的功能障碍能够引起较大的网络拓扑变化,可能需要为数据分组重新寻找路由以及网络重组。因此,节能和功率管理特别重要。正是由于这些原因,极其需要功率意识的 WSN 协议和算法。

在其他移动 Ad Hoc 网络中,功耗已经是一个重要设计因素,但不是首要考虑的因素,就是因为用户可以替换能量资源。相对于能量效率,设计重点更强调提供 QoS。但是,在 WSN 中,能量效率是一个重要性能指标,直接影响 WSN 寿命。对于根据特定应用适当设计的协议,需要合理平衡能量效率和其他性能指标(比如时延、吞吐量)。

在传感器场合中,传感器节点的主要任务就是检测事件,就地对数据进行迅速处理,然后发送数据。因此,将功耗分成三个方面的功耗:感知功耗、通信功耗、数据处理功耗。

前面对感知单元及其组成做了介绍。感知功耗随着应用的特性而变化。随机感知的功耗可能低于恒定事件监视的功耗。事件检测的复杂性对确定能耗起着关键作用。较高环境噪声可能引起严重恶化,以及增大检测复杂度。下面详细讨论数据通信功耗和数据处理功耗。

1. 通信功耗

在三个功耗中,传感器节点在数据通信方面消耗的能量最大,其中包括数据发送和数据接收的功耗。对于低辐射(约 0 dBm)短距离通信,发送能量开销和接收能量开销几乎相同。混频器、频率合成器、压控振荡器、锁相环(Phase Locked Loop, PLL),以及功率放大器,

全部都要消耗收发信机电路中的珍贵能量。在传感器节点的数据通信功耗中，不仅要考虑收发信机电路正常工作时的功耗，而且还要考虑收发信机电路启动时的功耗。启动时间长达数百毫秒，因此，不能忽略启动功耗。启动时间长的原因在于 PLL 花费的锁相时间。随着发送分组长度的减小，启动功耗开始对工作功耗占优势。结果，由于每次重新开启收发信机消耗很大一部分能量，因此导致收发信机开启和关闭低效率。

电台功耗计算公式 P_C [见参考文献[43]]如下

$$P_C = N_T[P_T(T_{on}+T_{st})+P_{out} \times T_{on}] + N_R[P_R(R_{on}+R_{st})]$$

式中， P_T 表示发射机消耗的功率； P_R 表示接收机消耗的功率； P_{out} 表示发射机的输出功率； T_{on} 表示发射机的工作时间； R_{on} 表示接收机的工作时间； T_{st} 表示发射机的启动时间； R_{st} 表示接收机的启动时间； N_T 表示单位时间内发射机开启的次数； N_R 表示单位时间内接收机开启的次数。 N_T 、 N_R 依赖传感器承担的具体任务以及所使用的 MAC 协议。 T_{on} 可以进一步表示为 $T_{on}=L/R$ ， L 表示分组的长度， R 表示数据速率。对于当前技术水平的低功率电台收发信机， P_T 、 P_R 的典型值均在 20 dBm 左右，而 P_{out} 接近 0 dBm^[34]。而 PicoRadio 追求的 P_C 目标值为 -20 dBm。

在设计小型、低成本、超低功率收发信机时，建议收发信机电路采用直接变换体系结构^[37]，估计功耗 P_T 、 P_R 比上面介绍的值至少小一个数量级。

2. 数据处理功耗

数据处理能耗比数据通信能耗小得多^[38]。假定瑞利衰落和四阶功率距离损耗，那么在 100 m 距离上发送 1 KB 信息的能耗约等于 100 万条指令每瓦每秒 (MIPS/W) 处理器执行 300 万条指令的能耗。因此，本地数据处理对于多跳 WSN 的功耗最小化非常关键。

因此，传感器节点必须具有内置计算能力，能够与其周围环境交互。再加上成本和体积约束条件，所以需要选择 CMOS 的微型处理器。但是，这限制了能量效率。一对 CMOS 晶体管在开关切换的时候都会消耗能量，切换功率与切换频率、装置电容（与面积有关）、电压波动范围成正比。因此，降低电压是降低活动状态下功耗的有效方法。电压动态调整方法^[51]的目的在于使处理器功率源和工作频率自适应工作载荷。微处理器在处理时变计算载荷时，在处理任务减轻期间降低工作频率会使功耗按线性下降，但是降低工作电压却会使增益按平方递增。另一方面，这样会影响处理器的峰值性能。不是总是需要峰值性能，所以可以得到较大的能量增益，因此，处理器的工作电压和工作频率可以动态自适应即时处理要求。在参考文献[44]中，作者提出了一个基于对以往工作载荷情况自适应滤波的工作载荷预测方案，分析了几种滤波方案。参考文献[38, 49, 71]讨论了其他的低功率 CPU 组织策略。

数据处理功耗 (P_P) 计算公式如下

$$P_P = CV_{dd}^2 f + V_{dd} I_0 e^{V_{dd}/n'V_T}$$

式中， C 表示总切换电容； V_{dd} 表示电压波动范围； f 表示切换频率；第二项表示泄漏电流造成的功耗^[44]。通过降低门限电压满足性能要求会增大门限泄漏电流。结合传感器节点中微处理器的低占空因数操作，有关功耗变得相当大^[43]。

可能需要一些其他电路用于编码和解码。在有些情形下，可能需要采用特定应用的集成电路。在上述各种节能方案中，包括已经讨论过的方案，WSN 算法和协议的设计都要受到相应功耗的影响。

参 考 文 献

- [1] C. E. Nishimura and D. M. Conlon. IUSS dual use: Monitoring whales and earthquakes using SOSUS. *Mar. Technol. Soc. J.*, Vol.27, No.4, pp.13–21, 1994.
- [2] Proceedings of the Distributed Sensor Nets Workshop. Pittsburgh,PA: Dept. Comput. Sci., Carnegie Mellon Univ., 1978.
- [3] R. F. Sproull and D. Cohen. High-level protocols. *Proc. IEEE*, Vol.66, pp.1371–1386, Nov. 1978.
- [4] P. Nii, E. Feigenbaum, J. Anton, and A. Rockmore, Signal-to-symbol transformation: HASP/SIAP case study. *AI Mag.*, Vol.3, pp.23–36, Spring 1982.
- [5] R. R. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Trans. Comput.*, Vol.29, pp.1104–1113, Dec. 1980.
- [6] V. Lesser and D. Corkill. Functionally accurate, cooperative distributed systems. *IEEE Trans. Syst., Man, Cybern.*, Vol. 11, pp.81–96, Jan./Feb. 1981.
- [7] R. B. Wesson, F. A. Hayes-Roth, J. W. Burge, C. Stasz, and C. A. Sunshine. Network structures for distributed situation assessment. *IEEE Trans. Syst., Man, Cybern.*, Vol. SMC-11, pp. 5–23, Jan./Feb.1981.
- [8] R. Rashid and G. Robertson. Accent: A communication oriented network operating system kernel. in *Proc. 8th Symp. Operating System Principles*, 1981, pp. 64–75.
- [9] R. Rashid, D. Julin, D. Orr, R. Sanzi, R. Baron, A. Forin, D. Golub, and M. Jones. Mach: A system software kernel. in *34th Computer Society Int. Conf. (COMPCON)*, San Francisco, CA, 1989.
- [10] C. Myers, A. Oppenheim, R. Davis, and W. Dove. Knowledgebased speech analysis and enhancement. presented at the *Int. Conf. Acoustics, Speech and Signal Processing*, San Diego, CA, 1984.
- [11] C.Y. Chong, S. Mori, and K. C. Chang. Distributed multitarget multisensor tracking. in *Multitarget Multisensor Tracking: Advanced Applications*, Y. Bar-Shalom, Ed. Norwood, MA: Artech House, 1990, pp. 247–295.
- [12] R. T. Lacoss. Distributed mixed sensor aircraft tracking. presented at the *Amer. Control Conf.*, Minneapolis, MN, 1987.
- [13] Distributed sensor networks. MIT Lincoln Laboratory, Lexington, MA, Rep. No. ESD-TR-88-175, 1986.
- [14] V. R. Lesser and D. D. Corkill. The distributed vehicle monitoring testbed: A tool for investigating distributed problem solving networks. *AI Mag.*, Vol. 4, No. 3, pp. 15–33, Fall 1983.
- [15] D. S. Alberts, J. J. Garska, and F. P. Stein. (1999) Network Centric Warfare: Developing and Leveraging Information Superiority [Online] Available: <http://www.dodccrp.org/NCW/ncw.html>.

- [16] (1995) The cooperative engagement capability. [Online] Available: <http://techdigest.jhuapl.edu/td1604/APLteam.pdf>.
- [17] Y. Carts-Powell. (2000, Apr.) Unattended ground sensors stop and analyze the roses. OE Rep. [Online] Available: <http://www.spie.org/web/oer/april/apr00/cover2.html>.
- [18] S. Kumar and D. Shepherd. SensIT: Sensor information technology for the warfighter. in Proc. 4th Int. Conf. on Information Fusion, 2001, pp. TuC1-3–TuC1-9.
- [19] J. Corella. Tactical automated security system (TASS): Air force expeditionary security. presented at the SPIE Conf. Unattended Ground Sensor Technologies and Applications, Orlando, FL, 2003.
- [20] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Mobile networking for smart dust. in Proc. ACM/IEEE Int. Conf. Mobile Computing and Networking (MobiCom), 1999, pp. 271–278.
- [21] P. Bauer, M. Sichitiu, R. Istepanian, K. Premaratne. The mobile patient: wireless distributed sensor networks for patient monitoring and care. Proceedings 2000 IEEE EMBS International Conference on Information Technology Applications in Biomedicine, 2000, pp. 17–21.
- [22] P. Bonnet, J. Gehrke, P. Seshadri. Querying the physical world. IEEE Personal Communications (October 2000) 10–15.
- [23] N. Bulusu, D. Estrin, L. Girod, J. Heidemann. Scalable coordination for wireless sensor networks: self-configuring localization systems, International Symposium on Communication Theory and Applications (ISCTA 2001). Ambleside, UK, July 2001.
- [24] A. Cerpa, J. Elson, M. Hamilton, J. Zhao. Habitat monitoring: application driver for wireless communications technology. ACM SIGCOMM'2000, Costa Rica, April 2001.
- [25] A. Chandrakasan, R. Amirtharajah, S. Cho, J. Goodman, G. Konduri, J. Kulik, W. Rabiner, A. Wang. Design considerations for distributed micro-sensor systems. Proceedings of the IEEE 1999 Custom Integrated Circuits Conference, San Diego, CA, May 1999, pp. 279–286.
- [26] S. Cho, A. Chandrakasan. Energy-efficient protocols for low duty cycle wireless microsensor. Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, Maui, HI Vol. 2 (2000), p. 10.
- [27] I.A. Essa. Ubiquitous sensing for smart and aware environments. IEEE Personal Communications (October 2000) 47–49.
- [28] D. Estrin, R. Govindan, J. Heidemann, S. Kumar. Next century challenges: scalable coordination in sensor networks. ACM MobiCom'99, Washington, USA, 1999, pp. 263–270.
- [29] G. Hoblos, M. Staroswiecki, A. Aitouche. Optimal design of fault tolerant sensor networks. IEEE International Conference on Control Applications, Anchorage, AK, September 2000, pp. 467–472.
- [30] P. Johnson et al.. Remote continuous physiological monitoring in the home. Journal of Telemed Telecare 2 (2)(1996) 107–113.
- [31] J.M. Kahn, R.H. Katz, K.S.J. Pister. Next century challenges: mobile networking for smart dust. Proceedings of the ACM MobiCom'99, Washington, USA, 1999, pp. 271–278.
- [32] R. Min, T. Furrer, A. Chandrakasan. Dynamic voltage scaling techniques for distributed

- microsensor networks. Proceedings of ACM MobiCom'95, August 1995.
- [33] J. Mirkovic, G.P. Venkataramani, S. Lu, L. Zhang. A self-organizing approach to data forwarding in large-scale sensor networks. IEEE International Conference on Communications ICC'01, Helsinki, Finland, June 2001.
 - [34] National Semiconductor Corporation. LMX3162 Single Chip Radio Transceiver. Evaluation Notes and Datasheet, March 2000.
 - [35] A. Perrig, R. Szewczyk, V. Wen, D. Culler, J.D. Tygar. SPINS: security protocols for sensor networks. Proceedings of ACM MobiCom'01, Rome, Italy, 2001, pp.189–199.
 - [36] E.M. Petriu, N.D. Georganas, D.C. Petriu, D. Makrakis, V.Z. Groza. Sensor-based information appliances. IEEE Instrumentation and Measurement Magazine (December 2000) 31–35.
 - [37] A. Porret, T. Melly, C.C. Enz, E.A. Vittoz. A low-power low-voltage transceiver architecture suitable for wireless distributed sensors network. IEEE International Symposium on Circuits and Systems'00, Geneva, Vol.1, 2000, pp.56–59.
 - [38] G.J. Pottie, W.J. Kaiser. Wireless integrated network sensors. Communications of the ACM 43 (5) (2000) 551–558.
 - [39] J. Rabaey, J. Ammer, J.L. da Silva Jr., D. Patel. PicoRadio: ad-hoc wireless networking of ubiquitous lowenergy sensor/monitor nodes. Proceedings of the IEEE Computer Society Annual Workshop on VLSI (WVLSI'00), Orlando, Florida, April 2000, pp.9–12.
 - [40] J.M. Rabaey, M.J. Ammer, J.L. da Silva Jr., D. Patel, S.Roundy. PicoRadio supports ad hoc ultra-low power wireless networking. IEEE Computer Magazine (2000)42–48.
 - [41] A. Savvides, C. Han, M. Srivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. Proceedings of ACM MobiCom'01, Rome, Italy, July 2001, pp.166–179.
 - [42] E. Shih, B.H. Calhoun, S. Cho, A. Chandrakasan. Energy efficient link layer for wireless microsensor networks. Proceedings IEEE Computer Society Workshop on VLSI 2001, Orlando, FL, April 2001, pp.16–21.
 - [43] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, A. Chandrakasan. Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks. Proceedings of ACM MobiCom'01, Rome, Italy, July 2001, pp.272–286.
 - [44] A. Sinha, A. Chandrakasan. Dynamic power management in wireless sensor networks. IEEE Design and Test of Computers, March/April 2001.
 - [45] S. Vardhan, M. Wilczynski, G. Pottie, W.J. Kaiser. Wireless integrated network sensors (WINS): distributed in situ sensing for mission and flight systems. IEEE Aerospace Conference, Vol.7, 2000, pp.459–463.
 - [46] A. Woo, D. Culler. A transmission control scheme for media access in sensor networks. Proceedings of ACM MobiCom'01, Rome, Italy, July 2001, pp.221–235.
 - [47] P. Gupta and P. R.Kumar. The capacity of wireless networks. IEEE Trans. Inform. Theory, vol.46, pp.388–404, Mar. 2000.
 - [48] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M.Srivastava. Coverage Problems in Wireless Ad-Hoc Sensor Network. Proc. IEEE INFOCOM '01, pp.1380–1387, 2001.

- [49] Xiang-Yang Li, Peng-Jun Wan and Ophir Frieder. Coverage in Wireless Ad Hoc Sensor Networks. Proc. IEEE Transactions on Computers, Vol.52, No.6, pp.1-111, June 2003.
- [50] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak. Exposure in Wireless Ad-Hoc Sensor Network. Proc. IEEE MOBICOM '01, pp.139-150, 2001.
- [51] S. Fortune. Voronoi Diagrams and Delaunay Triangulations. Computing in Euclidean Geometry, F.K. Hwang and D.-Z. Du, eds.,pp.193-233, Singapore: World Scientific, 1992.
- [52] K.J. Supowit. The Relative Neighborhood Graph, with an Application to Minimum Spanning Trees. J. ACM, Vol. 30, 1983.
- [53] K. Kar, S. Banerjee. Node placement for connected coverage in sensor networks. Proceedings of WiOpt : Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks 2003.
- [54] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, C.D. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. SenSys'03, pp28-39, 2003.
- [55] V. Raghunathan, C. Schurgers, S. Park, M.B. Srivastava. Energy-aware wireless microsensor networks. IEEE Signal Processing Magazine 19 (2002) 40–50.

第 2 章 无线传感器网络竞争类MAC协议

2.1 传感器媒介访问控制协议 (S-MAC)

传感器 MAC 协议 (Sensor-MAC, S-MAC) 是专门为 WSN 设计的 MAC 协议。尽管降低能耗是 S-MAC 协议的主要目标,但是通过综合运用时间安排协议和竞争协议, S-MAC 协议也能够达到良好的可扩展能力和碰撞回避能力。为了实现能量效率这个主要目标,需要确定导致能量低效使用的主要原因,以及确定能够降低能耗而需要做出哪些综合平衡。

2.1.1 能量浪费原因分析

导致能量浪费的主要原因分析如下:第一个原因是碰撞,一个发送分组被损坏后不得不被丢弃,随后分组重传增加了能耗,碰撞还导致时延的增大;第二个原因是分组旁听,即节点接收发送给其他节点的分组;第三个原因是控制分组开销,控制分组的发送和接收均需要消耗能量;最后一个原因,也是主要原因是空闲侦听,即侦听接收可能发送却还没有发送的分组。在很多 WSN 应用中尤其是如此。假如没有什么可侦听的,那么节点大部分时间处在空闲状态。但是,在许多 MAC 协议(比如 IEEE 802.11 Ad Hoc 方式、CDMA)中,节点必须侦听信道,以便接收可能的传输。测试结果表明空闲侦听消耗接收所需能量的 50%~100%。例如,Stemm 和 Katz 的测试结果是:能耗之比空闲:接收:发送=1:1.05:1.4, Digitan 无线局域网模块 (IEEE 802.11/2/2 Mb/s) 技术规范说明能耗之比空闲:接收:发送=1:2:2.5。大多数 WSN 设计成长时间工作,节点长时间处于空闲状态。因此,这种情况下空闲侦听是能量浪费的决定因素。

S-MAC 协议试图减轻上述所有原因造成的能量浪费。在交换过程中,允许每跳公平性和时延稍有一些下降。在 S-MAC 中采用的第一个技术是在多跳网络中建立低占空因数节点操作。通过周期性地使节点进入休眠状态,减少空闲侦听。在休眠状态,电台被完全关闭。在传统无线数据网络协议(比如 IEEE 802.11)中,带宽利用率是一个重点考虑的问题,节点通常完全工作在活动方式。切换到低占空因数方式是每个节点的一种选择方式,通常在一个节点空闲时间长的时候才会这样做。但是,在 S-MAC 中,低占空因数方式是所有节点的默认操作方式;节点只有在网络中存在传输流量之时才会重新变成活动节点。为了降低控制开销和减小时延, S-MAC 引入了相邻节点间可协调休眠机制。

根据所执行的应用,时延可能重要,也可能不重要。在诸如监视或者监测之类的应用中,节点长时间保持警惕,但是大部分时间处于非活动状态,直到检测到某种事物或者事件才会变成活动状态。这种应用常常能够容忍一些额外的消息发送传输时延,这是因为网络速度通常是所观测物理目标速度的几个数量级。所观测目标速度限制网络必需的反应速度。在没有所观测事件期间,网络中的数据流量通常非常小。次要重要的时延不重要,增大时延换取能

量的节省。因此，S-MAC 使节点周期性地进入休眠状态，否则节点处于空闲状态。这种设计降低了能耗，但是增大了时延，这是因为发送节点必须等待接收节点醒来之后才能够发送其数据。S-MAC 协议的自适应侦听技术能够大幅度减小这种时延。

在传统的无线语音或者无线数据网络中，每个用户需要同等机会和同样多时间访问媒介，即为它们自己的应用进行分组的发送或者接收。因此，每跳 MAC 层的公平是一个重要问题。但是，在 WSN 中，所有节点相互协作，一起完成一个共同任务。一个节点在任何时刻都有可能比其他节点多出许多的数据需要发送。在这种情况下，公平性的重要性不及应用级的性能，后者不能下降。S-MAC 协议重复引入消息分片传输机制来高效发送长消息。消息分片传输机制的基本思想是将长消息分成多个小片，然后按照突发方式发送各个小片。其结果是具有较多数据需要发送的节点获得较多时间访问媒介。从每跳 MAC 层来看，这对于只有一些短分组需要发送的节点来说是不公平的。但是，消息分片传输机制降低了控制开销和避免了旁听，因而节省了能量。

WSN 的一个重要特性是网内数据处理。相对于将原始数据尽数发送给端节点，网络数据处理通常能够降低能耗。数据累积之类的技术能够降低流量，而联合信号处理能够降低流量和提高感知质量。网内数据处理要求对消息进行存储转发处理。一条消息是一个有意义的、且节点能够处理（均化或者滤波等）的数据单元。一条消息可能很长，由许多小片组成。在这种情况下，支持分片级公平性的 MAC 协议实际上增大了消息级的时延。而消息分片传输机制正好相反：综合平衡分片级公平性，减小了消息级的时延。

2.1.2 S-MAC协议概述

S-MAC 协议包括几种方法，用于降低空闲侦听、碰撞、旁听、控制开销造成的能量浪费。在描述 S-MAC 协议组成之前，首先概括对有关 WSN 及其应用所做的假设。

WSN 由大量节点组成，采用短距离多跳通信节能。大部分通信是对等方式节点间的通信，而不是与单个中心节点（如基站等）的通信。网内处理对网络寿命至关重要，这就意味着按照存储转发方式将数据作为完整的消息来处理。来自多个源节点的分组或者分片交错存储只会引起总时延的增大。最后，假设应用具有长空闲周期，并且能够容忍的时延高达网络消息通信时间的数量级。

1. 周期性的侦听与休眠

如上所述，在很多 WSN 应用中，假如没有发生感知事件，则节点长时间处于空闲状态。假定空闲期间数据速率非常低，没必要使节点一直保持在侦听状态。S-MAC 使节点周期性地进入休眠状态，从而缩短侦听时间。

周期性侦听与休眠的基本方案如图 2-1（a）所示。每个节点休眠一定时间，然后醒来侦听信道，检查是否有其他节点需要与其通信。在休眠期间，节点关掉其电台，设置随后打开其电台的定时器。

将一个完整的侦听和休眠循环称为一个帧。根据物理层和 MAC 层参数，比如信道带宽、竞争窗口大小，侦听间隔通常是固定的。占空因数定义为侦听间隔与帧长之比率。可以根据不同应用要求改变休眠间隔，应用要求实际上是改变占空因数。为了简单起见，这些参数数值对于所有节点都是相同的。

所有节点自由选择自己的侦听/休眠时间安排。但是，为了降低控制开销，宁愿相邻节点相互同步，也就是说，相邻节点在相同时刻侦听、在相同时刻进入休眠状态。应该注意到：在多跳网络中，并不是所有相邻节点都能够相互同步。假如两个相邻节点 A 和 B 必须分别与不同的节点 C 和 D 同步，则 A 和 B 可能有不同的时间安排，如图 2-1 (b) 所示。

节点周期性给其直接相邻节点广播一个同步 (SYNC) 分组，实现时间安排的交换。节点在其所安排的侦听时间与其相邻节点通信，因此确保所有节点即使在有不同时间安排条件下也仍然能够通信。例如，在图 2-1 (b) 中，假如节点 A 想与节点 B 通信，那么节点 A 需要一直等到节点 B 侦听为止。节点发送 SYNC 分组的时间周期称为同步周期。

S-MAC 协议的一个特点是将节点构成一个平面对等拓扑。S-MAC 不同于分群协议，不需要通过群首进行协调。在 S-MAC 中，节点围绕公共时间安排建立虚拟分群，但是与对等节点直接通信。这种松散协调的一个优点是拓扑变化的适应能力强于分群法。

S-MAC 的缺点是周期性休眠引起时延的增大。而且，这种时延可能逐跳累加。稍后将介绍一种能够大幅度减小这种时延的技术，即自适应侦听技术。

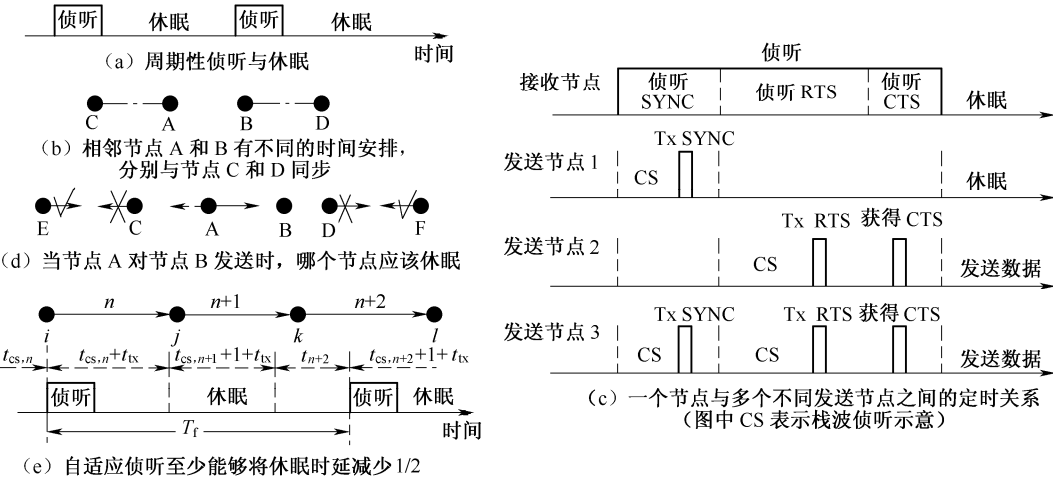


图 2-1 S-MAC 协议描述示意图

2. 碰撞回避

假如多个相邻节点在同一时刻想与同一个节点通信，那么这些相邻节点试图在这个节点开始侦听之时进行发送。因此这些相邻节点需要竞争媒介。在竞争协议中，IEEE 802.11 通过载波侦听、碰撞回避、控制分组交互来解决传输媒介竞争问题。S-MAC 协议采用类似的规程，包括解决隐含终端问题的虚拟载波侦听和物理载波侦听，以及 RTS/CTS 交互。

在所发送的每个分组中有一个持续时间组成域，用于说明本分组剩余的传输时间。一个节点接收到传递给另一个节点的一个分组后，就能够根据该分组中持续时间域的数值而决定自己需要保持多长时间的静默，并将该值记录到一个叫做网络分配矢量 (Network Allocation Vector, NAV) 的变量中，然后设置相应的定时器。每当启动定时器时，节点递减其 NAV，直到 NAV 等于零为止。节点在开始发送之前首先检查其 NAV：若其 NAV 不等于零，节点判定媒介忙，叫做虚拟载波侦听。

物理载波侦听就是物理层侦听信道上可能的传输。载波侦听时间是竞争窗口内的一个随

机时间，以便避免碰撞和饥饿状态。假如虚拟载波侦听和物理载波侦听都判定媒介空闲，才判定媒介空闲。

所有节点在发送之前进行载波侦听。节点若没有获取传输媒介，则进入休眠状态，在接收节点重新空闲和侦听信道时退出休眠状态而进入活动状态。广播分组不采用 RTS/CTS 发送。单目标分组在发送节点和接收节点之间按照 RTS/CTS/DATA/ACK 顺序进行发送。在 RTS 和 CTS 交互成功后，发送节点和接收节点采用其正常休眠时间进行数据分组传输，在完成数据传输之前不遵循其休眠时间安排。

由于在每个侦听期间采用了低占空因数操作和竞争机制，所以 S-MAC 协议有效解决了空闲侦听和碰撞造成的能量浪费问题。下面详细介绍相邻节点间周期性休眠的协调问题，然后进一步介绍能够降低旁听和控制开销造成的能量浪费的两种技术。

2.1.3 休眠的协调

周期性休眠有效地降低了空闲侦听造成的能量浪费。在 S-MAC 协议中，各个节点相互协调其休眠时间安排，而不是各自随机地休眠。本节详述所有节点必须遵循的时间安排的建立与维护规程，以及介绍一种能够降低周期性休眠给每个节点造成的时延的技术。

1. 时间安排的选择和维护

每个节点在开始其周期性侦听和休眠之前需要选择一个时间安排，并与其相邻节点交换这个时间安排。每个节点维护一张时间安排表，用于存储其所知道的所有相邻节点的时间安排。节点遵循以下步骤选择其时间安排和建立其时间安排表：

① 节点首先侦听信道一段固定时间，这段时间至少等于同步周期。节点若是没有侦听到其他节点发送来的时间安排，则立即选择自己的时间安排，并开始执行该时间安排。同时，节点试图广播一个 SYNC 分组，以便将该时间安排通知给其相邻节点。SYNC 分组广播遵从标准竞争规程。载波侦听时间随机化能够减少 SYNC 分组的碰撞。

② 节点若是在选择或者广播其自己的时间安排之前接收到一个相邻节点的时间安排，则遵从这个时间安排，并将自己的时间安排设置成与这个时间安排一样。然后试图广播通知这个时间安排作为其下一个已安排的侦听时间。

③ 假如节点在选择和广播其自己的时间安排之后接收到一个不同的时间安排，则存在两种情况。该节点若是没有其他相邻节点，则抛弃其当前时间安排，遵从所接收到的这个新时间安排。该节点若是已经在与一个或者多个相邻节点共同遵从从一个时间安排，则采用这两个时间安排，在这两个时间安排的侦听期间退出休眠状态而进入活动状态。

为了说明这个算法，考虑一个网络：所有网络节点都能够相互接收到对方的发送。首先启动的节点首先选出一个时间安排，做出的 SYNC 分组广播将使所有对等节点同步到这个时间安排上。假如两个或者更多节点在相同时刻首先启动，则这些节点将在相同时刻完成初始侦听，并且将独立选出一个相同的时间安排。不论哪个节点首先发出其 SYNC 分组（赢得竞争），该节点都将与剩余节点同步。

但是，在多跳网络中，两个节点若是不能接收到对方的发送，则可能独自分配时间安排。在这种情况下，处于两个时间安排边界上的那些节点将采用两个时间安排。例如，图 2-1 (b) 中的节点 A 和 B 将在两个时间安排的侦听期间退出休眠状态而进入活动状态。这样，当一个

边界节点发送一个广播分组的时候，该节点只需将该广播分组发送一次。这种方法的缺点是边界节点的休眠时间较少、因而消耗的能量多于其他节点。

另外一种选择是让边界节点只选择一个时间安排——只选择首先接收到的那个时间安排。边界节点知道有一些相邻节点遵从另外一个时间安排，所以仍然能够与这些相邻节点通信。但是，对于广播，边界节点需要对两个不同时间安排做出两次发送。这种方法的优点是边界节点具有与其他节点同样简单的周期性侦听和休眠模式。

接收到多个时间安排的情况非常少，这是因为每个节点在选择一个独立时间安排之前努力遵从现有的时间安排。但是，新节点可能找不到某个现行相邻节点，其原因是多方面的。这个相邻节点发送的 SYNC 分组可能由于碰撞或者干扰而被破坏，可能由于媒介忙而推迟了 SYNC 分组的发送。假如新节点处在两个时间安排的边界上，并且假如这两个时间安排不重叠，那么这个新节点可能只能找到第一个相邻节点。

当两个相邻节点遵从完全不同的时间安排时，为了防止这两个相邻节点总是找不到对方，S-MAC 协议采用周期性相邻节点搜寻机制，即每个节点周期性侦听整个同步周期。节点执行相邻节点寻找的频度依赖该节点所具有的相邻节点数量。一个节点若是没有相邻节点，则比具有多个相邻节点时更加主动、更加频繁地执行相邻节点寻找。由于在相邻节点寻找期间的能量开销很高，所以不应该太频繁地执行相邻节点寻找。在后面介绍的 S-MAC 协议实现中，同步周期 10 s，假如一个节点至少有一个相邻节点，那么该节点每隔 2 min 执行一次相邻节点寻找。

2. 同步维护

由于相邻节点相互协调其休眠时间安排，所以每个节点的时钟漂移可能造成同步错误。S-MAC 协议采用两种技术来提高抗同步错误的能力。首先，所交换的所有时戳是相对的，不是绝对的。第二，侦听周期比时钟漂移速率大许多。例如，0.5 s 的侦听时间比典型的时钟漂移速率大 10^4 倍。对比时隙极短的 TDMA 协议，S-MAC 协议的时间同步宽松得多。

尽管长侦听时间能够容忍相当大的时钟漂移速率，但是相邻节点仍然需要周期性相互使用对方的时间安排来更新其时间安排，以便防止出现长时间的时钟漂移。同步周期可以相当长。测试床上的节点测试结果表明：两个节点之间的时钟漂移不能大于 0.3 ms/s。

如上所述，时间安排通过发送 SYNC 分组来完成。SYNC 分组非常短，包含发送节点的地址及其下一次休眠时间。下一次休眠时间是发送节点 SYNC 分组开始发送时刻的相对时间。接收节点接收到 SYNC 分组后，将其中的下一次休眠时间减去该 SYNC 分组的传输时间，得到新的时间，用这个新时间调整其定时器。

为了使节点接收 SYNC 分组和数据分组，将其侦听间隔分成两个部分：第一个部分用于接收 SYNC 分组，第二部分用于接收数据分组，如图 2-1 (c) 所示。每个部分均有一个竞争窗口，竞争窗口包含许多用于进行载波侦听的时隙。例如，假如发送节点需要发送一个 SYNC 分组，那么发送节点在接收节点开始侦听之时开始进行载波侦听。发送节点随机选择一个时隙来完成载波侦听。若在该时隙结束之时没有侦听到发送，则发送节点赢得媒介竞争，接着开始发送其 SYNC 分组。数据分组发送遵循同样的规程。

图 2-1 (c) 表示发送节点给接收节点发送时三种可能情形的定时关系：发送节点 1 只发送 SYNC 分组；发送节点 2 只发送单目标分组；发送节点 3 既发送 SYNC 分组，也发送数据分组。

3. 自适应侦听

在轻流量载荷时，周期性侦听和休眠能够大幅度地减少空闲侦听所消耗的时间。但是，当确实出现感知事件的时候，要求感知数据在网路中传递不要经历太大的时延。当每个节点严格地遵从其休眠时间安排时，每个转发跳可能增加时延，其平均值与帧长成正比。因此，给 S-MAC 协议引入一个机制，将这种情形下的节点状态从低占空因数工作方式切换到比较活动的工作方式。

S-MAC 协议建议采用一个叫做自适应侦听的重要技术来改善多跳网络中每个节点周期性休眠造成的时延性能。基本思想是使旁听其相邻节点发送(理想情况下只是 RTS 或者 CTS)的节点在该发送结束之时苏醒一段短时间。这样，假如该节点是下一个转发跳节点，那么其相邻节点就能立即将数据发送给该节点，而不需要等到到达其安排的侦听时间。该节点在自适应侦听期间若是没有接收到任何信息，则退回到休眠状态，直到到达下一个安排的侦听时刻为止。

现在考虑图 2-1 (c) 的定时关系，下一个转发跳节点若是发送节点的相邻节点，则会接收到 RTS 分组。下一个转发跳节点若是接收节点的相邻节点，则会接收到接收节点发送的 CTS 分组。因此，发送节点和接收节点的相邻节点均会分别从 RTS 和 CTS 分组中的持续时间域中获得本次传输将持续的时间长度。所以，当传输时间结束时，发送节点和接收节点的相邻节点就能够自动苏醒。

自适应侦听的时间间隔不包括 SYNC 分组的传输时间，但是正常侦听时间间隔却包括 SYNC 分组的传输时间，如图 2-1 (c) 所示。SYNC 分组只是在所安排的侦听时间发送，以确保能够被所有相邻节点所接收。为了给 SYNC 分组优先权，假如从完成前一次传输之时到所安排的正常侦听时间之间的时间小于自适应侦听时间间隔，那么不执行自适应侦听和发送。

应该注意到并不是所有下一个转发跳节点都能够旁听到前一次发送的分组，特别是前一次发送自适应启动(即不是在所安排的侦听时间启动)的时候尤其如此。所以，发送节点若是在自适应侦听期间发送一个 RTS 分组而启动发送，则可能接收不到 CTS 应答。在这种情况下，发送节点正好退回到休眠状态，然后在下一个正常侦听时间又重新尝试发送。

2.1.4 避免旁听与消息分片传输

碰撞回避是 MAC 协议的一个基本任务。S-MAC 协议采用竞争类 MAC 方案来实现碰撞回避。一个节点发送的任意分组将被其所有相邻节点所接收，即使该分组的接收节点是其中的某个相邻节点。碰撞导致竞争类协议的能量效率低于 TDMA 协议。

1. 避免旁听

在 IEEE 802.11 协议中，每个节点连续侦听其相邻节点的所有发送，以便执行有效的虚拟载波侦听。结果是每个节点旁听到许多不是发送给自己的分组。这样浪费大量的能量，特别是在节点密度高、流量载荷重的时候能量浪费更加严重。

由于受到 PAMAS 协议的启示，S-MAC 协议在旁听到干扰节点发送的 RTS 分组或者 CTS 分组后，使干扰节点进入休眠状态，从而尽量避免旁听。由于数据分组通常比控制分组大得

多，所以这种方法能够防止相邻节点旁听长数据分组及其随后的 ACK 分组。现在的问题是：若正在进行一个传输，那么哪些节点应该进入休眠状态？

在图 2-1 (d) 中，节点 A、B、C、D、E、F 构成一个多跳网络，每个节点能够旁听到其直接相邻节点的发送。假定节点 A 正在对节点 B 发送一个数据分组。那么在这个分组传输期间，哪些剩余节点应该进入休眠状态？

碰撞是在接收节点一方发生的。显然，节点 D 应该休眠，这是因为节点 D 的发送干扰节点 B 的接收。节点 E 和 F 不会产生干扰，所以不必休眠。节点 C 应该休眠吗？节点 C 离节点 B 两跳远，其发送不会干扰节点 B 的接收，所以可以自由地向其相邻节点（如节点 E）发送。但是，节点 C 不能接收到节点 E 的发送（如 CTS 分组或者数据分组），这是因为节点 E 的发送与节点 A 的发送在节点 C 上发生碰撞。所以节点 C 的发送只是浪费能量。而且，节点 A 给节点 B 发送后，需要等待节点 B 发送的 ACK 应答分组，节点 C 的发送可能损坏这个 ACK 应答分组。总之，发送节点和接收节点的所有直接相邻节点在旁听到 RTS 分组或者 CTS 分组后，都应该进入休眠状态，直到当前传输结束为止，见图 2-1 (d) 中的“×”。

每个节点维护一个 NAV，用于记录其相邻节点的活动情况。一个节点接收到一个发送给其他节点的分组后，利用该分组中持续时间域数值更新其 NAV。非零 NAV 表示其相邻区域内还在进行发送。每当启动 NAV 定时器的时候，将 NAV 减 1。因此，若是一个节点的 NAV 不等于零，那么这个节点应该休眠，避免进行旁听；当其 NAV 等于零时，则节点苏醒。

但是在有些情况下确实需要旁听。有些算法可能依靠旁听来收集相邻区域信息，用于网络监视、可靠路由或者分布式查询。假如需要旁听，则可以配置 S-MAC 协议，允许进行应用特定的旁听。但是，不需要旁听的算法可能更适合能量有限的网络。例如，S-MAC 协议采用直接数据应答，而不是采用隐含应答。

2. 消息分片传输

本节将从能量和时延两个方面同时考虑长消息的高效发送。一条消息就是数据的有意义、相关单元的集合。接收节点通常需要接收到所有数据单元后才能够进行网内数据处理或者数据累积。

将长消息作为一个分组来发送的缺点是：假如第一次传输只发生少数几比特误码，那么长分组重传开销高。但是，假如我们将长消息分片成多个独立的短分组，那么必须付出的代价是控制开销高、时延大，这是因为发送每个独立分组时竞争媒介都采用了 RTS 分组和 CTS 分组。

S-MAC 协议的处理方法是将长消息分成多个小片，按照突发方式发送各个小片。RTS 分组和 CTS 分组只使用一次。预留媒介用于发送所有的小片。每当发送一个数据小片时，发送节点等待接收节点回送的 ACK 分组。发送节点若是没有接收到接收节点回送的 ACK 分组，则将预留的发送时间增加一个数据小片的传输时间，然后立即重传当前数据小片。

如前一样，所有分组都包含一个持续时间域，用于记录发送所有剩余数据分片和 ACK 所需要的时间（注意这里与前面的不同）。一个相邻节点旁听到一个 RTS 分组或者 CTS 分组，就进入休眠状态，休眠时间长度等于发送所有剩余数据分片所需要的时间。

每个数据分片和 ACK 分组都有一个持续时间域。假如在一个传输期间有一个节点苏醒或者有一个新节点加入网中，那么这个节点可以适当进入休眠状态，而不管这个节点是否为发送节点或者接收节点的相邻节点。假如发送节点由于分片丢失或者误码而延长传输时间，

那么正在休眠的相邻节点不会立即知道延长的时间，但是苏醒后通过所旁听到的多传输的数据分片或者 ACK 分组就能够获知延长的时间。

在每个数据分片之后采用 ACK 是为了防止隐含终端，即防止在传输期间某个相邻节点苏醒或者一个新节点入网。假如一个节点只是接收节点而不是发送节点的相邻节点，那么这个节点不会旁听到发送节点发送的数据分片。假如接收节点不经常发送 ACK 分组，那么新节点可能根据其载波侦听错误地推断媒介是空闲的。假如新节点启动发送，则当前传输可能在接收节点上受到破坏。

值得注意的是：IEEE 802.11 协议也支持分片传输机制。在 IEEE 802.11 协议中，RTS 分组和 CTS 分组只是为第一个数据分片和第一个 ACK 分组预留媒介。第一个数据分片和 ACK 分组为第二个数据分片和 ACK 分组预留媒介，以此类推。每个相邻节点接收到一个数据分片和 ACK 分组后，就知道随后还要发送一个数据分片，所以必须继续侦听信道，直到所有数据分片发送完毕为止。此外，对于能量有限的节点，所有相邻节点进行旁听浪费许多能量。

IEEE 802.11 协议增强了公平性。发送节点若是没有接收到所发送数据分片的 ACK 分组，则必须退出发送，重新竞争媒介，这样其他节点才有机会发送。假如接收节点确实需要接收到整条消息后才能够开始对其进行处理，那么这种处理方法的时延可能很长。而 S-MAC 协议的消息分片传输机制与此大不相同，采取延长传输时间、重传当前数据分片处理方法，减少了竞争，缩短了时延。S-MAC 协议对每条消息的延长传输时间做了限制，以防接收节点在传输期间真正失效或者丢失连接。但是，对于 WSN，目标是应用级性能，而不是每个节点的公平性。

2.1.5 时延分析

本节分析 MAC 协议的多跳时延，定量分析 S-MAC 协议周期性休眠引入的时延。对于在多跳网络中传递的一个分组，其每跳传递的时延包括如下：

① 载波侦听时延 (Carrier Sense Delay) 为发送节点进行载波侦听时花费的时间，载波侦听时延大小由竞争窗口大小决定。

② 退避时延 (Backoff Delay) 为节点侦听到另外一个发送或者发生碰撞而导致载波侦听失败时所产生的时延。

③ 传输时延 (Transmission Delay) 由信道带宽以及所采用的分组长度和编码方案决定。

④ 传播时延 (Propagation Delay) 由发送节点和接收节点之间的距离决定。在 WSN 中，节点距离通常很短，因此通常可以忽略传播时延。

⑤ 处理时延 (Processing Delay) 为节点在将所收分组转发到下一个转发跳之前对该分组所作处理而花费的时间。处理时延主要依赖节点的计算能力，以及网内数据处理算法的效率。

⑥ 排队时延 (Queueing Delay) 依赖于流量载荷。在重载荷情形下，排队时延变成一个支配因素。

上述时延是采用竞争类 MAC 协议的多跳网络的固有时延。这些因素对于 S-MAC 协议和 IEEE 802.11 类协议也是相同的。S-MAC 协议中的另外一种时延是每个节点周期性休眠造成的时延。发送节点获得一个分组需要发送时，必须等到接收节点苏醒。这种时延是接收节点休眠造成的，所以将这种时延称为休眠时延。

下面分析流量载荷极轻情形下不同 MAC 协议的时延。流量载荷极轻的含义假定为只有

一个分组在网络中传递，所以没有排队时延和退避时延。另外假定传播时延和处理时延可以忽略不计。在这种情形下，只考虑载波侦听时延、传输时延、休眠时延。

假定分组从源节点经过 N 个转发跳传递到达中心节点。每个转发跳的载波侦听时延是随机的，第 n 个转发跳的载波侦听时延表示为 $t_{CS, n}$ 。 $t_{CS, n}$ 的平均值由竞争窗口大小 c_{CS} 决定。分组长度固定，则传输时延 t_{tx} 固定。

首先分析没有休眠机制的 MAC 协议。节点接收到一个分组时，立即启动载波侦听，试图将该分组转发到下一跳。第 n 个转发跳的平均时延为 $t_{CS, n} + t_{tx}$ 。全部 N 跳的总时延为

$$D(N) = \sum_{n=1}^N (t_{CS, n} + t_{tx}) \quad (2-1)$$

所以没有休眠机制的 MAC 协议的 N 跳平均时延为

$$E[D(N)] = N(t_{CS} + t_{tx}) \quad (2-2)$$

式 (2-2) 表明：在没有休眠机制的 MAC 协议中，多跳时延随着转发跳数的递增而线性增大。直线的斜率等于平均载波侦听时间与分组传输时间之和。

现在分析 S-MAC 协议。S-MAC 协议的每个转发跳都有休眠时延。第 n 个转发跳的休眠时延表示为 $t_{S, n}$ 。为了简单起见，假定路径上的所有节点遵循相同的时间安排。一个帧就是一个完整的侦听和休眠周期，其长度表示为 T_f 。侦听间隔固定，所以可以通过调整休眠间隔来改变帧长度。为了反映极低的占空因数，即 $\leq 10\%$ ，假定 T_f 取大值， $T_f > t_{tx}$ 。在第 n 个转发跳的时延为

$$D_n = t_{S, n} + t_{CS, n} + t_{tx} \quad (2-3)$$

在没有自适应侦听的 S-MAC 协议中，竞争（载波侦听）只是在每帧的开始（即每个节点开始侦听的时刻）才启动。一个节点接收到一帧中一个分组后，必须等到下一个转发跳节点苏醒，下一个转发跳节点苏醒时刻就是下一帧开始时刻。表示为

$$T_f = t_{CS, n-1} + t_{tx} + t_{S, n} \quad (2-4)$$

在第 n 个转发跳的休眠时延为

$$t_{S, n} = T_f - (t_{CS, n-1} + t_{tx}) \quad (2-5)$$

将式 (2-5) 代入式 (2-3)，得到第 n 个转发跳的时延为

$$D_n = T_f + t_{CS, n} - t_{CS, n-1} \quad (2-6)$$

第 1 个转发跳 ($n=1$) 是例外，这是因为源节点可以在任何时间产生一帧中一个分组。所以第 1 个转发跳的休眠时延 $t_{S, 1}$ 是一个随机变量，其值位于 $(0, T_f)$ 之间。假设 $t_{S, 1}$ 均匀分布于 $(0, T_f)$ 之间， $t_{S, 1}$ 的均值等于 $T_f/2$ 。与式 (2-6) 比较，得到一个分组传递 N 个转发跳后的总时延为

$$D(N) = D_1 + \sum_{n=2}^N D_n = t_{S, 1} + t_{CS, 1} + t_{tx} + \sum_{n=2}^N (T_f + t_{CS, n} - t_{CS, n-1}) = t_{S, 1} + (N-1)T_f + t_{CS, N} + t_{tx} \quad (2-7)$$

所以，没有自适应侦听的 S-MAC 协议的 N 跳平均时延为

$$E[D(N)] = E[t_{S, 1} + (N-1)T_f + t_{CS, N} + t_{tx}] = NT_f - T_f/2 + t_{CS} + t_{tx} \quad (2-8)$$

式 (2-8) 表明：在 S-MAC 协议中，当每个节点严格遵循其休眠时间安排时，多跳时延也随着转发跳数的增大而线性增大，直线的斜率等于帧长度 T_f 。与式 (2-2) 比较，由于占空因数非常低，所以通常 T_f 比 $(t_{CS} + t_{tx})$ 大许多。因此，周期性休眠给每跳都增加了新的时延。

现在分析具有自适应侦听的 S-MAC 协议。图 2-1 (e) 表示一个多跳网络的一部分，即三个转发跳表示为 n 到 $(n+2)$ 。假定所有节点遵循同一个休眠时间安排。

假定节点 i 首先等待节点 j 在其所安排的正常侦听时刻苏醒，接着启动载波侦听，以便在该时刻发送数据。第 n 个转发跳的时延仍然表示为式 (2-3)。

在节点 i 和 j 交换 RTS/CTS 分组期间，下一个转发跳节点 k 也在进行侦听以及旁听节点 j 发送的 CTS 分组。所以节点 k 知道节点 i 对节点 j 的发送何时结束。前一次传输一旦结束，那么自适应侦听机制立即唤醒节点 k ，同时让节点 j 启动载波侦听，侦听对节点 k 的发送。因此，第 $(n+1)$ 个转发跳的时延为

$$D_n = t_{CS,n+1} + t_{tx} \quad (2-9)$$

对比前一个转发跳的时延，本跳没有休眠时延。假如帧长 T_f 大于 $(t_{CS} + t_{CS,n+1} + 2t_{tx})$ ，那么分组正好在一帧内传递通过了两个转发跳。假定在下面的分析中保持这个条件，这是因为假定了 T_f 比 t_{tx} 大许多。

另一方面，节点 l 离节点 j 两跳远，可能旁听不到节点 j 发送的 CTS 分组；但是节点 k 能够旁听到节点 j 发送的 CTS 分组。在这种情形下，节点 l 不能在节点 i 对节点 j 的发送结束时苏醒。当节点 j 在正常休眠时间开始对节点 k 发送时，节点 l 仍然处在休眠状态而意识不到节点 j 对节点 k 的发送。因此，节点 l 不能在节点 j 对节点 k 的发送结束时苏醒。节点 k 必须等到到达节点 l 的正常侦听时间才能开始其发送。第 $(n+2)$ 个转发跳的时延仍然表示为式 (2-3)。

因此，在具有自适应侦听的 S-MAC 协议中，其他任意转发跳都存在休眠时延。 N 个转发跳的时延为

$$D(N) = t_{S,1} + t_{CS,1} + t_{tx} + t_{CS,2} + t_{tx} + t_{CS,3} + t_{tx} + \cdots + t_{CS,n-1} + t_{tx} + t_{CS,n} + t_{tx} \quad (2-10)$$

注意到[见图 2-1 (e)]

$$T_f = t_{CS,n} + t_{tx} + t_{CS,n+1} + t_{tx} + t_{S,n+2} \quad (2-11)$$

所以式 (2-10) 可以简化为

$$D(N) = t_{S,1} + (N/2 - 1)T_f + t_{CS,n-1} + t_{CS,n} + 2t_{tx} \quad (2-12)$$

因此，具有自适应侦听的 S-MAC 协议的 N 跳平均时延为

$$E[D(N)] = T_f/2 + (N/2 - 1)T_f + 2t_{CS} + 2t_{tx} = NT_f/2 - T_f/2 + 2t_{CS} + 2t_{tx} \quad (2-13)$$

从式 (2-13) 中可以看到：在具有自适应侦听的 S-MAC 协议中，多跳时延仍然随着转发跳数的增大而线性增大，但是直线的斜率等于帧长度 $T_f/2$ 。对比没有自适应侦听的 S-MAC 协议的多跳时延[见式 (2-8)]，加入自适应侦听机制后，S-MAC 的多跳时延减少 1/2。

式 (2-13) 是在只有一跳相邻节点能够相互接收到对方发送，但是两跳相邻节点却不能相互接收到对方发送的假设条件下得到的。在实际中，通常这个假设条件不成立。无线传播理论及其测试结果表明：接收信号功率 P_r 随着距离 d 的增大而下降

$$P_r \propto P_t d^\beta \quad (2-14)$$

式中， P_t 表示发射功率； β 表示依赖环境的常数，其取值范围在 2~5 之间。显然，传输距离不会突然停止在某个传输距离上。

考虑图 2-1 (e)，假如节点 k 能够可靠接收节点 j 的发送，即正确接收率 95% 以上，那么节点 l 仍然能够以较高概率接收节点 j 发送的 CTS 分组（特别是非常短的 CTS 分组）。假如两跳相邻节点以 20%~30% 概率相互接收对方的发送，那么由于有些两跳相邻节点也能够参与自适应侦听，所以总时延可以进一步减小。

2.1.6 S-MAC协议实现

实现 S-MAC 协议的目的是为了证明 S-MAC 协议的效能，以及将其与其他没有 S-MAC 协议所有节能特性的协议进行比较。采用 UCB Mote 传感器作为开发平台和测试床。UCB Mote 传感器运行 TinyOS。

在 Mica Mote 传感器上实现 S-MAC 协议。Mica Mote 传感器包含一个 Atmel ATmega128L 微处理器，该处理器包含 128 KB 闪存和 4 KB 数据存储器。Mica Mote 传感器配置有 RFM TR3000 电台收发信机、一副匹配鞭状天线。调制方案是幅移键控（Amplitude Shift Keying，ASK）。电台接收、发送、休眠有三种工作方式，其耗能分别为 14.4 mW、36 mW、15 μW。

S-MAC 协议实现不是基于 TinyOS 发布版的标准通信栈，而是实现一个协议栈，这个协议栈包含一些 S-MAC 协议的关键特征。首先，这个协议栈采用分层化体系结构。各层提供标准接口和服务，因此可以并行开发各层的协议。这个协议栈将物理层功能和 MAC 层功能明确分开。物理层直接控制电台，提供上层 API，以便使电台进入不同状态，即休眠、空闲、发送、接收。物理层完成符号检测、信道编码和解码、字节缓存、CRC 校验，提供载波侦听功能，但是完全由 MAC 层控制。

这个协议栈采用嵌套分组头结构，以便于分组定义。每层自由定义其分组类型，对上层下载来的分组添加本层的分组头。一个组件在定义自己的分组格式或者分组头时，必须在其第一个域中填写其直接相邻的下一协议层的分组头。这样。每个分组缓存器包含来自所有协议层的所有分组头。因此，避免了跨层的存储器复制。

表 2-1 列出一些重要参数。采用曼彻斯特码作为信道编码方案。曼彻斯特码是强壮的 DC 平衡码，其开销为 1:2，也就是说每个数据比特经过曼彻斯特编码后变成两比特。根据对协议和硬件特性的理解选择这些参数。

S-MAC 协议的实现允许用户在编译时选择不同选项，将 S-MAC 配置成不同的工作方式。下面给出一些重要选项：

- ① 占空因数选择。这个选项允许用户选择 S-MAC 的各种占空因数，其范围在 1%~99% 之间。
- ② 完全活动方式。这个选项完全关闭周期性休眠，主要用于性能比较研究。
- ③ 关闭自适应侦听。低占空因数方式下的默认方式就是自适应侦听。这个选项关闭自适应侦听，每个节点严格遵循其侦听时间安排。

节点的其他硬件也可以进入休眠，包括 CPU。

表 2-1 在 Mica Mote 传感器上实现的 S-MAC 协议的参数

信 道 带 宽	20 kb/s
信道编码	曼彻斯特码
控制分组长度	10 B
数据分组长度	最多 250 B
MAC 头长度	8 B
占空因数	1%~99%
侦听间隔	115 ms
SYNC 竞争窗口	15 个时隙
数据竞争窗口	31 个时隙

2.1.7 S-MAC协议的性能

实验的目的是为了揭示 S-MAC 协议的能量、时延、吞吐量之间的基本平衡关系。比较所实现的不同 MAC 协议模块的性能。为了便于测试多跳网络中传递的多条消息，在每个节点的应用层增加一个消息队列，用于缓存输出消息。

1. 能耗

为了测量电台的能耗，测量电台在不同工作方式下花费的时间：休眠、空闲、接收、发送。然后计算每种工作方式的能耗：将消耗时间乘以在该工作方式下工作所需要的能量。间接测量能量是因为在体积小、低功率 Mica Mote 传感器上直接观测当前能耗很困难。在不同流量载荷下比较各种 MAC 协议的能耗。

(1) 两跳网络的能耗

在 Rene Mote 传感器上实现的 MAC 协议上进行实验。拓扑是两跳网络，两个源节点，两个中心节点，见图 2-2 (e)。源节点 A 发送的分组经过节点 C 到达中心节点 D，源节点 B 发送的分组也经过节点 C 到达中心节点 E。

通过改变消息到达间隔周期来改变流量载荷。假如消息到达间隔周期为 5 s，那么每个源节点每隔 5 s 产生一条消息。在这项实验中，消息到达间隔周期变化范围为 1~10 s。对于 1 s 到达间隔周期的最高速率，无线信道带宽低而几乎被全部利用。对于每种流量模式，针对不同 MAC 协议做 10 次独立实验。

在每个实验中，每个源节点周期性产生 10 条消息，每条消息被分成 TinyOS 支持的 10 个小数据分组（每个 40 B）。因此，在每个实验中，200 个 TinyOS 数据分组从源节点传递到其中心节点。测量每个节点在发送固定数量数据时的电台能耗。对于每种 MAC 协议，完成数据发送所需要的真正时间是不同的。

在 IEEE 802.11 类似协议中，采用突发方式发送每条消息的各个分片，即在发送该条消息的第一个分片时才使用 RTS 分组和 CTS 分组。没有测试无分片机制的 IEEE 802.11 类似协议，这种协议将每个分片当做一个独立分组来处理，每个分片都使用 RTS 分组和 CTS 分组。显然，这种协议的能耗比具有分片机制的 IEEE 802.11 类似协议高得多。在 S-MAC 协议中，采用消息分片传输机制，并且总是采用突发方式发送每条消息的各个分片。在包含周期性休眠的 S-MAC 协议中，按照占空因数 50% 配置各个节点。

图 2-2 (a) 表示对源节点 A 和 B 测试得到的平均能耗。当消息到达间隔周期小于 4 s 时，流量重。在重流量条件下，IEEE 802.11 MAC 协议的能耗比 S-MAC 协议高出 2 倍以上。因为很少发生空闲侦听，所以周期性休眠的节能非常有限。S-MAC 协议主要通过旁听回避和长消息高效发送来实现节能。

当消息到达间隔周期大于 4 s 时，流量变轻。在轻流量条件下，完整 S-MAC 协议能量性能最好，远胜于 IEEE 802.11 MAC 协议。包含旁听回避的消息分片传输的能量性能也优于 IEEE 802.11 MAC 协议。但是，如图 2-2 (a) 所示，当空闲侦听对总能耗起主导作用时，周期性休眠对节能起着关键作用。

对比 IEEE 802.11 MAC 协议，包含旁听回避的消息分片传输在所有流量条件下几乎节省同样多的能量。其原因在于相邻节点 A、B、C 之间的旁听回避。在所有流量条件下，这几个相邻节点发送相同数量的分组。

(2) 多跳网络的能耗

在 Mica Mote 传感器上实现的 MAC 协议上进行实验。拓扑是直线网络, 11 个节点, 见图 2-2 (f)。各个节点按照最低功率发送, 并且放置在 1 m 的空间内。第一个节点是源节点, 最后一个节点是中心节点。

如前所述, 通过改变源节点消息到达间隔周期来改变流量载荷。消息到达间隔周期变化范围为 0~10 s, 消息到达间隔周期等于 0 s 表示源节点在相同时刻产生和排队缓存所有分组。在每种流量模式条件下, 做 5 次独立实验。在每个实验中, 源节点发送 20 条消息, 每条消息 100 B, 所有消息不分片。

比较 S-MAC 协议的三种不同操作方式。第一种操作方式是 10% 占空因数, 没有自适应侦听; 第二种方式是 10% 占空因数, 自适应侦听; 第三种方式是完全活动方式, 彻底关闭周期性休眠。因为周期性侦听间隔为 115 ms, 所以 10% 占空因数相当于帧长 1.15 s。

图 2-2 (b) 表示固定数量数据从源节点发出, 经过网络传递最终到达中心节点时整个网络中所测得的电台能耗。结果与前面两跳网络的实验结果一致。在多跳网络中, 尤其是在轻流量载荷条件下, 周期性休眠 S-MAC 协议的节能优势比无周期性休眠 MAC 协议强得多。

通过比较两种以 10% 占空因数操作的 MAC 协议模块, 可以看到自适应侦听 MAC 协议的能量效率优于无自适应侦听 MAC 协议, 特别是重流量载荷下优势更加明显。主要原因是自适应侦听大幅度减少了固定数量数据在网络中传递所需要的总时间。

2. 端到端时延

由于 S-MAC 协议为了节能而对时延做了折中, 所以在多跳网络中, 由于每个节点的周期性休眠, 所以可能时延较大。S-MAC 协议的自适应侦听就是要使这种额外时延最小。为了定量时延和测试自适应侦听带来的效果, 仍然采用图 2-2 (f) 所示的 10 跳网络拓扑来测试 S-MAC 协议的端到端时延。

考虑两种极端流量情形: 最轻流量载荷和最重流量载荷。在最轻流量载荷条件下, 源节点在中心节点接收到前一条消息之后才产生下一条消息。为此, 在中心节点附近布置一个协调节点。协调节点接收到消息后, 以最大功率直接给源节点发送消息。由于流量载荷轻, 所以每个节点没有排队时延。对比没有休眠的 MAC 协议, 额外时延只是由每个节点周期性休眠产生的。在最重流量载荷条件下, 所有消息在同一时刻由源节点产生和排队缓存。所以每个节点的排队时延最大, 其中包括源节点。在这两种极端情形下, 开始测量每条消息从其在源节点产生之时开始的时延。

在每个实验中, 源节点产生 20 条消息, 每条消息 100 B。所有消息不分片。在最轻流量载荷下, 分组产生时间在一帧时间内均匀分布。在最轻流量载荷和在最重流量载荷条件下重复十次实验。同样对 S-MAC 协议的三种不同操作方式进行测试。

图 2-2 (c) 给出最轻流量载荷条件下所测得的每跳平均消息时延。在 S-MAC 协议的所有三种不同操作方式中, 时延随着转发跳数的增多而线性增大。但是, 无自适应侦听、10% 占空因数的 S-MAC 协议的时延大于另外两种操作方式的 S-MAC 协议。其原因是每条消息必须每跳等待一个休眠周期。

相比之下, 自适应侦听 S-MAC 协议的时延非常接近无周期性休眠的 MAC 协议, 这是因为自适应侦听常常允许 S-MAC 协议立即将消息发送给下一个转发跳。但是, 这个时延仍然没有完全活动方式 MAC 协议的最短时延那样短。正如前面所述, 自适应侦听不能保证每个转发跳都立即发送。假如一个节点发送了一个 RTS 分组, 但是没有接收到预定接收节点回送

的 CTS 分组，那么该节点必须等到下一个周期。图 2-2 (c) 表明：自适应侦听 S-MAC 协议的时延约为完全活动方式 MAC 协议时延的 2 倍（第一个转发跳或者两跳除外）。从实验中还观察到：对于任何一种占空因数，其时延的变化比完全活动方式 MAC 协议大得多，并且随着转发跳数的增大而增大。时延变化大的原因是有些消息可能错过某些节点的休眠周期。

图 2-2 (d) 表示最重流量载荷条件下的每跳平均时延。此时，无自适应侦听、10% 占空因数的 S-MAC 协议的时延仍然最大。采用自适应侦听后，S-MAC 协议的时延接近完全活动方式 MAC 协议的时延，但是前者仍然约等于后者的 2 倍。

两种低占空因数操作方式之间第一个转发跳的时延差异大的原因在于源节点的排队时延。没有采用自适应侦听和传输，在一个循环周期发送一条消息，所以最后一条消息必须至少等待 19 个循环周期。随着消息的向前传递，后面转发跳的排队时延较小。总的结果是：没有自适应侦听的低占空因数操作方式的斜率低于图 2-2 (c) 所示的斜率。

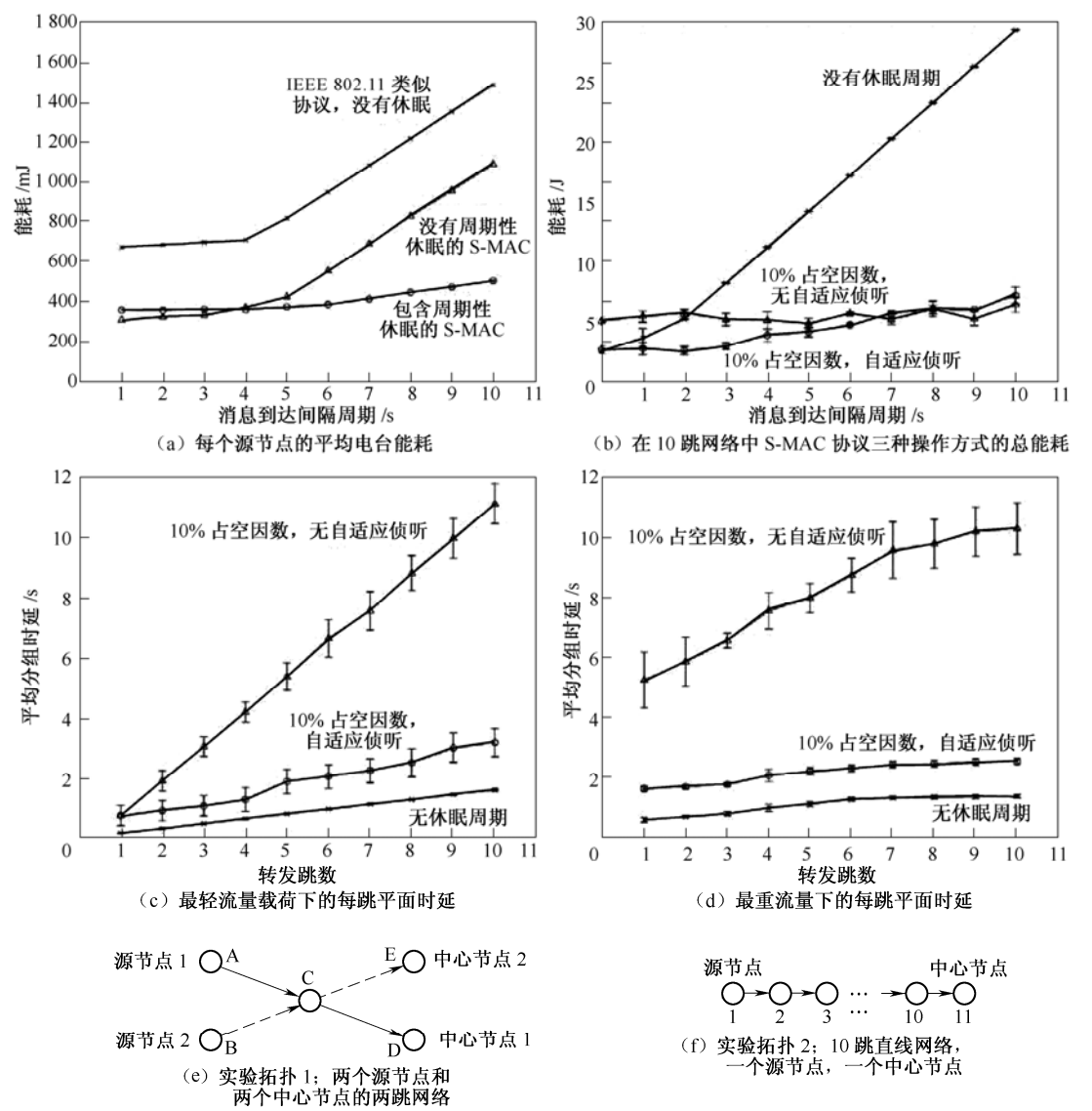


图 2-2 S-MAC 协议的性能

自适应侦听、低占空因数操作方式总是能够在这样重的流量载荷条件下发送数据，所以其斜率相同于完全活动方式的斜率，这个效果也减弱了时延的变化。

2.2 超时MAC协议（T-MAC）

S-MAC 协议是单信道竞争协议，其基本思想是将时间划分成非常大的帧。每个帧有两个组成部分：活动部分和休眠部分。在休眠部分，节点关闭其电台，以节省能量。在活动部分，节点与其相邻节点通信，发送在休眠期间排队缓存的消息，如图 2-3（a）所示。由于所有消息都是集中在活动部分发送的，没有在整个帧（即消息之间的间隔时间）上进行传播（发送和接收），因此减少了空闲侦听的能耗。

S-MAC 协议需要一定程度的同步，但是没有 TDMA 类协议同步要求那么严格，时间刻度大得多。通常情况下，在一个一秒帧中活动部分长 200 ms，时钟漂移 500 μ s 不会引起时间同步问题。

S-MAC 协议必须以能量为代价来改善吞吐量和时延。由于只有帧的活动部分才用于通信，所以吞吐量下降。由于可能在休眠期间产生消息事件，这样消息排队等待下一个活动时间，所以时延增大。

荷兰代夫特理工大学（Delft University of Technology）研发的超时 MAC 协议（Timeout MAC, T-MAC）协议改进了 S-MAC 协议的能量使用方法，即在每个活动周期开始时设置极短的侦听窗口。完成活动周期的 SYNC 周期后，采用一个小窗口来发送或者接收 RTS 分组或者 CTS 分组。假如本活动周期内没有活动，则节点返回到休眠状态。

2.2.1 T-MAC协议概述

能耗是 T-MAC 协议设计的主要准则。除了空闲侦听引起的能耗问题外，其他形式的能耗如下：

- ① 碰撞：假如两个节点在相同时刻发送并且相互干扰，那么所发送的分组将被破坏。因此，发送和接收使用的能量被浪费。
- ② 协议开销：大多数协议需要交换控制分组。因为控制分组不包含应用数据，用于发送和接收控制分组的所有能量都是开销。
- ③ 旁听：由于无线信道是共享媒介，所以节点可以接收到不是发送给自己的分组，然后可以关闭自己的电台。

相对于空闲侦听浪费的能量，特别是在消息发送不频繁的时候，其他能耗源相对不重要。例如，考虑一个 WSN 应用：要求节点以平均一个分组每秒的速度与其相邻节点交换消息。消息相当短，其发送时间小于 5 ms。因此，每个节点平均每秒花费 5 ms 发送一条消息、花费 5 ms 接收一条消息，剩余的 990 ms 用于信道侦听、但是没有侦听到任何信息。所以，电台在 99% 的时间内没有做任何事情。假如实际发送和接收时间随着两个因数增大——碰撞和开销，那么空闲侦听时间只能从 99% 降到 98%。

固定占空因数的解决方法（比如 S-MAC 协议）尽管能够减少侦听时间，但不是最佳解决方法。S-MAC 协议有两个重要参数：一是受到时延要求和缓存器大小限制的总帧时间，二是活动时间。活动时间主要依赖消息速率，可以小得让节点能够在活动时间内发送完所有消息。

问题是：尽管时延要求和缓存器大小通常是固定的，但是消息速率通常是变化的。假如无论如何重要消息不能丢失、而不重要消息不该发送，那么所配置的节点活动时间必须能够处理最重期望载荷。只要载荷低于最重期望载荷，那么就没有最佳使用活动时间，能量就会浪费在空闲侦听上。

T-MAC 协议的实现思想是：按照长度可变的突发方式发送所有消息，在突发发送之间休眠，从而减少空闲侦听。为了在可变载荷条件下维护最佳活动时间，动态确定活动时间的长度。凭直觉法结束活动时间，旁听不到信息时就认为时间到了。

2.2.2 T-MAC基本协议

图 2-3 (b) 给出了 T-MAC 协议的基本方案。每个节点周期性苏醒，与其相邻节点通信，然后又进入休眠状态，等到下一个帧。同时新消息排队等待。节点按照发送请求 (Request-To-Send, RTS)、允许发送 (Clear-To-Send, CTS)、数据、应答 (ACK) 规程相互通信，这个规程提供碰撞回避和可靠传输。IEEE 802.11 标准采用了这个规程。

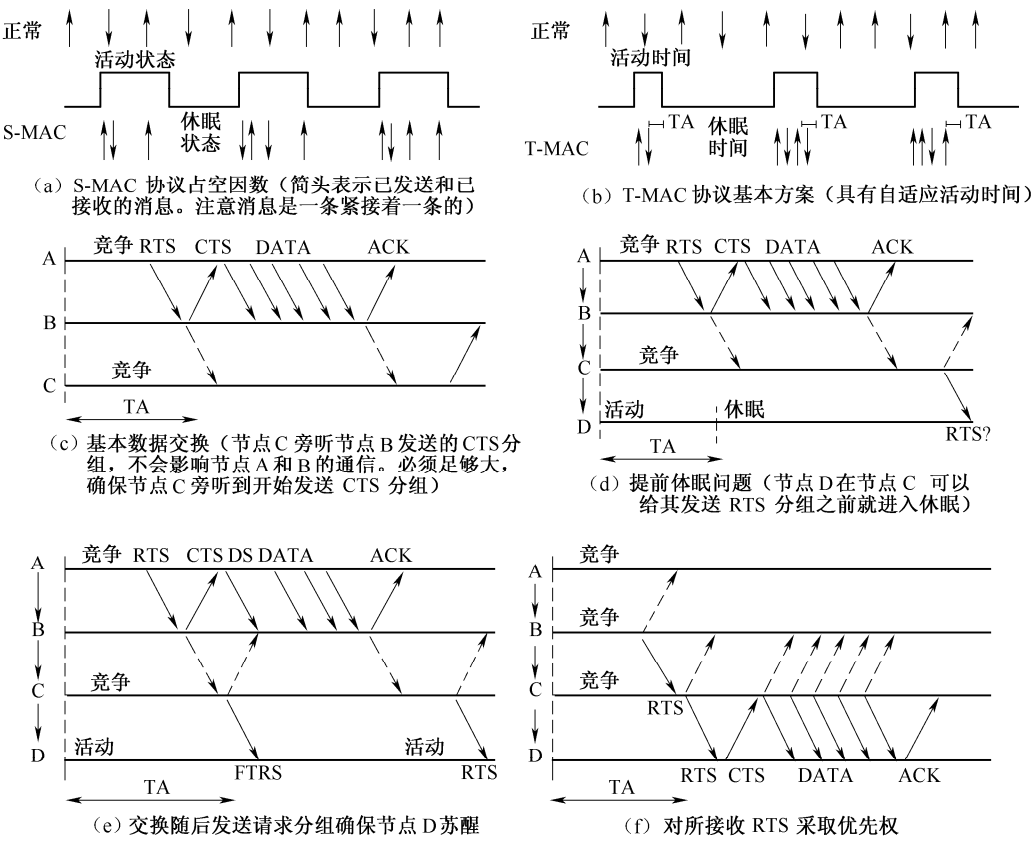


图 2-3 T-MAC 协议描述示意图

节点只要处在活动期间，就保持侦听信道以及可能进行发送。当在时间 TA 内没有发生活动事件时，就结束活动周期。活动事件是：

- 启动周期性帧定时器；
- 从信道上接收到任何数据；
- 侦听到信道上通信（根据电台的接收信号强度指示器（Received Signal Strength Indication, RSSI)), 比如在碰撞期间；
- 节点结束自己数据分组或者 ACK 的发送；
- 通过旁听到先前的 RTS 和 CTS 分组获知一个相邻节点的数据交换已经结束。

节点不在活动期间就休眠。因此 TA 决定每帧的最小侦听时间。

T-MAC 协议的超时方案是以突发方式在帧开始时进行所有通信。由于活动期间之间的消息必须排队缓存，所以缓存器容量决定最大帧时间的上限值。

2.2.3 分群与同步

T-MAC 协议的帧同步受到虚拟分群的启示，比如 S-MAC 协议中的虚拟分群。一个节点苏醒后开始等待和侦听信道。假如在一段时间内没有侦听到任何信息，则选择一个帧传输时间安排，发送一个 SYNC 分组，SYNC 分组包含到达下一个帧开始时的时间。假如节点在启动期间侦听到另一个节点发送的 SYNC 分组，则遵循这个 SYNC 分组中的传输时间安排，据此发送自己的 SYNC 分组。

节点隔一段时间重发自己的 SYNC 分组，还必须偶尔侦听一个完整帧，这样才能够检测到不同传输时间安排的存在。这就允许新的移动节点适应现有的节点组。

假如一个节点有一个不同于另一个节点的传输时间安排，那么该节点必须适应两个传输时间安排，而且还必须按照自己的传输时间安排给另外那个节点发送 SYNC 分组，以便让另外那个节点也知道存在另外一个不同的传输时间安排。适应两个传输时间安排意味着这个节点在两个帧开始时有一个活动事件。

节点必须只在自己活动时间开始时发送数据。具有相同传输时间安排的两个相邻节点，以及已经适应了另外一个传输时间安排的相邻节点在自己活动时间开始时刻苏醒。假如一个节点在其一个相邻节点帧的开始时刻开始发送，那么可能向另外一个休眠节点发送。注意：这个方案只需进行一次广播发送。

S-MAC 协议描述的虚拟分群同步方案要求节点按照一个相同传输时间安排构成分群，但是不要求全网节点都遵循这个传输时间安排。这就能够高效广播，不需要维护单个相邻节点的信息。

虚拟分群技术易于实现。由于活动时间相互重叠，所以保持多个活动时间固定不变的传输时间安排比较复杂。

2.2.4 RTS操作与TA选择

1. 固定竞争间隔

在竞争类协议中，比如 IEEE 802.11，节点检测到碰撞后，在竞争期间随机等待一段时间。其间只有在信道干净的时候，节点才会重新开始发送。通常采用退避方案：流量较高时，竞争间隔随着增大。退避方案降低重载荷时的碰撞概率，同时提供轻载荷的最小时延。

在 T-MAC 协议中，每个节点在帧开始时按照突发方式发送其排队缓存的消息。在突发发送期间，信道处于饱和状态：以最大速率发送消息。节点每当发送一个 RTS 分组时可能需要激烈竞争信道。因为载荷非常重且又不变化，所以增大竞争间隔是没有效果的。因此，T-MAC 协议在固定竞争间隔内等待和侦听一段随机时间后才开始发送 RTS 分组。固定竞争间隔可调整。即使还没有发生碰撞，也总是使用固定竞争间隔。

2. RTS重传

一个节点发送一个 RTS 分组后没有接收到相应的 CTS 分组，则已经发生下列事件之一：

- ① 接收节点由于碰撞没有接收到这个 RTS 分组；
- ② 接收节点由于旁听到 RTS 或者 CTS 而被禁止应答；
- ③ 接收节点正在休眠中。

发送节点在间隔时间 TA 内没有接收到应答，则可以进入休眠。但是，这样对于事件①和②是错误的：将会出现发送节点进入休眠、而接收节点仍然苏醒的情况。由于即使在帧的第一条消息也可能会发生这种情况，所以吞吐量急剧下降（在实验中确实如此）。

因此，假如接收节点没有应答，则发送节点应该重发 RTS 分组。假如两次重发 RTS 分组后仍然没有接收节点的应答，则发送节点放弃发送而进入休眠。

3. TA的确定

一个节点的相邻节点仍然在通信，这个节点可能是相邻节点随后发送消息的接收节点，因此不应该进入休眠。接收到相邻节点开始发送 RTS 分组或者 CTS 分组足够触发一个被更新的间隔时间 TA。

一个节点不在传输覆盖范围内，因而旁听不到启动与其相邻节点通信的 RTS 分组，因此，间隔时间 TA 必须足够大，确保至少接收到 CTS 分组开始部分，见图 2-3 (c)。根据这个观察结果，得到间隔时间 TA 的长度下限值： $TA > C + R + T$ ，其中，C 表示竞争间隔的长度，R 表示 RTS 分组的长度，T 表示来回时间（RTS 分组发送结束时刻与 CTS 分组开始发送时刻之间的最短时间）。在稍后介绍的 T-MAC 协议实验中采用 $TA = 1.5 \times (C + R + T)$ ，这已被证明是合适的。TA 越大，能耗就越大。

2.2.5 避免旁听

S-MAC 协议介绍了一种思想：节点旁听到发送给其他节点的 RTS 分组或者 CTS 分组后进入休眠。由于此时一个节点被禁止发送，所以该节点可以不参与任何通信，并且也可以关闭电台，节省能量。

通常，避免旁听是个好想法，也是 T-MAC 协议的一个选项。但是，通过实验已经观察到：作为副作用，碰撞开销变得较高：节点在休眠时可能丢失其他 RTS 分组和 CTS 分组、而在苏醒时可能干扰一些通信。其结果是，最大吞吐量下降，这是因为短分组占 25%。因此，尽管避免旁听节省能量，但是需要保证最大吞吐量（常常这样要求）时则不能采用避免旁听。

2.2.6 不对称通信

仿真实验揭示了采用 T-MAC 协议的一个问题：网络流量大多是单向的，就像多节点到中心节点通信模式那样。图 2-3 (d) 简化了这个问题。图 2-3 (d) 中每个节点（节点 A、B、C、D）与其相邻节点构成一个蜂窝。消息自上往下传递，所以节点 A 只给节点 B 发送，节点 B 只给节点 C 发送，节点 C 只给节点 D 发送。现在考虑节点 C。每当节点 C 需要给节点 D 发送消息的时候，节点 C 必须竞争信道，可能放弃对节点 B（通过接收到 RTS 分组）的竞争或者放弃对节点 A（间接地，通过旁听到节点 B 发送的 CTS 分组）的竞争。

假如节点 C 由于旁听到节点 B 发送的 RTS 分组而释放信道竞争，那么节点 C 应答 CTS 分组，这个 CTS 分组可以被节点 D 所接收到。在这种情况下，节点 D 在节点 C 和 B 的通信结束之时苏醒。但是，假如节点 C 由于旁听到节点 B 发送给节点 A 的 CTS 分组而释放信道竞争，那么节点 C 必须保持静默。因为节点 D 不知道节点 A 和 B 正在通信，所以节点 D 将结束其活动时间，进入休眠。节点 C 只有在下一帧开始的时候才有新机会给节点 D 发送。

因此，对于节点 C 需要发送给节点 D 的每个分组，节点 C 可能发送成功、也可能发送失败（通过释放与节点 A 的竞争）。这两种事件等概率发生。发送失败意味着帧结束、以及节点 C 不能再发送分组。因此可以计算出：在这种简单结构中，节点 C 有 50% 的概率在每个帧中给节点 D 发送单个分组，有 25% 的概率在每个帧中给节点 D 发送两个分组（节点 C 必须成功发送两次）和其他附加信息。

将这种观测结果称为提前休眠问题，其理由在于节点在其相邻节点还有消息发送给自己的时候进入休眠。在多节点到中心节点通信模式中，提前休眠问题导致 T-MAC 协议的总吞吐量下降到低于传统协议和 S-MAC 协议的最大吞吐量的一半。在后面的 T-MAC 协议实验中，已经在强活动网络部分的边界上遇到了这个问题。在任何不对称通信模式中均可能发生这个问题。T-MAC 协议采用如下两种方法来解决这个问题。

1. 随后发送请求法（FRTS）

第一个解决方案称为随后发送请求（Future Request-To-Send, FRTS）方案，其思想是：节点让另外一个节点知道自己还有消息发送给它，但是被禁止使用媒介。FRTS 解决方法的工作原理如下：一个节点旁听到发送给另外一个节点的 CTS 分组，则可以立即发送一个 FRTS 分组，见图 2-3 (e) 中的节点 C。FRTS 分组包含阻碍数据通信的长度（该信息包含在 CTS 分组中）。节点若是正好在该 CTS 分组之后检测到通信、或者由于前面的 RTS 分组或者 CTS 分组而被禁止发送，则一定不能发送 FRTS 分组。

一个节点接收到 FRTS 分组后，知道自己是随后发送的 RTS 分组的接收节点，并且在该 RTS 分组发送之时必须苏醒。该节点根据 FRTS 分组中的定时信息就能够确定苏醒时刻。否则，FRTS 分组将会干扰 CTS 分组之后的数据分组，所以在发送 FRTS 分组期间必须推迟数据分组的发送。为了防止任何其他节点在 FRTS 分组发送期间争夺信道，最初发送 RTS 分组的节点[见图 2-3 (e) 中的节点 A]发送一个短数据发送（Data-Send, DS）分组，发送完 DS 分组后必须立即发送正常的数据分组。

由于 FRTS、DS 的分组长度相同，所以 FRTS 分组会碰撞 DS 分组，但是不会碰撞其后的数据分组。DS 分组丢失，但是不会出现问题：DS 分组不包含任何有用信息。

FRTS 解决方法要起作用，必须增大 TA，使其包含一个控制分组（CTS）的长度，如图 2-3 (e) 所示。实现 FRTS 特性使单向通信模式的最大吞吐量提高约 75%。但是，由于 FRTS 和 DS 分组开销稍高，所以能耗也稍微提高。只有单向通信模式载荷重得合适，才可能需要使用 FRTS 分组。载荷轻，则所交换的分组数量少，因此所增加的开销也少。

2. 全缓存优先法

第二个解决方案称为全缓存优先法。当一个节点的发送缓存器/路由缓存器快要全满的时候，该节点可以使发送优先于接收。这就意味着一个节点接收到一个发送给自己的 RTS 分组后，立即给另一个节点发送一个自己的 RTS 分组，而不是像正常情况那样应答 CTS 分组，如图 2-3 (f) 所示。有两个效果，一个效果是这个节点根据旁听到的信道竞争 RTS 分组而有效赢取信道，因而平均具有较高机会发送自己的消息。在图 2-3 (f) 中，节点 C 在与节点 B 的信道竞争中失败后可以给节点 D 发送。因此，发生提前休眠问题的概率较低。第二个效果是全缓存优先法在网络中引入了一定程度的流量控制，这对多节点到中心节点通信模式是有利的。在图 2-3 (f) 中，节点 B 被禁止发送，直到节点 C 有足够缓存容量时才被允许发送。

但是，全缓存优先法不利于重载荷，在重载荷情形下通信不是单向的，所以采用全缓存优先法时必须非常仔细。当全向通信模式下的所有节点都采用全缓存优先法时，碰撞机会迅速增大。因此，T-MAC 协议使用一个门限：一个节点在竞争中至少失败两次的时候才可以只使用全缓存优先法。在后面的 T-MAC 协议实验中，门限保证全向通信模式的性能，同时仍然提高了单向通信模式的最大吞吐量。

2.2.7 T-MAC的性能

通过仿真实验比较三个协议：CSMA、S-MAC、T-MAC。考虑与 CSMA 比较的理由是将 CSMA 协议当做一种“最差情形”：CSMA 根本没有任何节能特征；节点不在发送时，则将其电台设置在接收方式下。考虑与 S-MAC 协议比较的理由是：S-MAC 协议像 T-MAC 协议一样，是为 WSN 设计的，采用带内信令。

1. 仿真建立与仿真参数

利用离散事件仿真包 OMNeT 建立无线传感器节点（EYES）逼真模型。该模型具有与 EYES 节点相同的时钟分辨率和精度、电台收发转换时间、电台苏醒时间、传输比特率。模型的能耗以实际节点能耗为基础：休眠 20 μ A，接收 4 mA，发送一个直流平衡信号 10 mA。

运用这些模型节点建立一个分布在 10×10 栅格内的 100 节点网络。设置的电台传输范围满足非边沿节点均有 8 个相邻节点。这个理想设置不是大多数 WSN 的真实设置，但是能够为 T-MAC 协议仿真需求提供服务，比如，不需要知道产生流量的每个节点的精确位置。

对于多节点到中心节点通信模式，采用随机最短路径路由法：对于每条消息，列出其传递的可能下一个转发跳。下一个转发跳到达最终目的节点的路径短于从发送节点到达最终目的节点的路径，那么这个转发跳符合要求。从这些下一个转发跳中随机选出一个。因此，消息朝正确方向传递，但是每次不会采用相同路径传递。这种路由法不需要交换任何控制消息：节点自动确定下一个转发跳。由于 T-MAC 协议采用带内信令——CTS 分组和 ACK 分组保持

直接相邻节点处于苏醒状态，所以多跳消息至少是每帧传递两个转发跳。

在仿真实验中，S-MAC 协议的实验配置是：帧长 1 s，活动时间在 75~915 ms 范围内可变；T-MAC 协议的实验配置是：帧长 610 ms (20 000 石英晶体单位)，间隔时间 TA 为 15 ms。T-MAC 协议的可选机制包括旁听回避、全缓存优先法、FRTS 法。下面将会说明每个实验中这些选项的最佳组合。

由于吞吐量不完全重要，所以图中曲线结束点位于被正确接收的消息低于 90% 时的点上。在多跳通信模式下，比如多节点到中心节点通信模式，意味着全部消息中的 90% 最终传递到达中心节点。对于 CSMA 协议没有 90% 的消息到达限制，因为 CSMA 没有内置重传机制，所以 CSMA 的可靠性差得多。

消息长度就是数据载荷长度，不包括 MAC 分组头：CSMA 的分组头为 4 B，S-MAC 和 T-MAC 的分组头均为 6 B。

2. 同类本地单目标传输能耗

在第一个实验中，节点随机地给其一个相邻节点发送分组，分组载荷为 20 B 或者 100 B。尽管这不是实际通信模式，但是可以作为一种“基本情况”。对于 T-MAC 协议，采用旁听回避机制，但是不采用 FRTS 机制和全缓存优先机制。实验结果如图 2-4 (a) 所示。

从图 2-4 (a) 中可以明显看到 CSMA 协议没有节能功能，其空闲时的能耗为 4 mA (电台接收方式下的工作电流)，并且随着发送消息的增多而稍微递增。

对于 S-MAC 协议，其不同曲线对应不同长度的活动时间。图中描绘了一条连接每种载荷下能耗最低的 S-MAC 协议曲线的曲线。所以，S-MAC 协议在这条曲线上是可调的，以便提供 90% 以上的吞吐量、同时在每种载荷下能耗又尽可能低。下面将只说明 S-MAC 协议的类似可调曲线。

同类实验对于 S-MAC 协议是最佳情形，这是因为载荷在时间和位置上都是恒定的。T-MAC 协议的性能表现（每种已调载荷）与 S-MAC 协议相同。

T-MAC 协议的能耗低于 S-MAC 协议的原因在于实验中 S-MAC 协议的活动时间是有限的几个离散数据。为了获得每种载荷的最佳参数值，S-MAC 协议需要进行复杂地调试，就像在实际使用中那样。但是，T-MAC 协议的自适应特征却不需要直接调试。

3. 多节点到中心节点通信模式下的能耗

在实验中，各个节点给位于网络角落处的中心节点发送消息。采用一种最短路径算法逐跳转发消息。不进行数据累积。对于 T-MAC 协议，采用旁听回避机制、全缓存优先机制、以及 FRTS 机制。实验结果如图 2-4 (b) 所示。

从图 2-4 (b) 中看到 T-MAC 协议的能耗低于 S-MAC 协议。希望达到这个结果，这是因为在多节点到中心节点通信模式中载荷随着节点位置的变化而变化：中心节点邻域中流量相对较重。这也说明了多节点到中心节点通信模式中本地单目标传输的绝对载荷轻得多（低 10 倍）：为了不拥塞网络（中心节点周围），中心节点的输入消息处理速度限制各个节点所能够产生的消息载荷。

正如同类本地单目标实验结果一样，T-MAC 协议的最大吞吐量低于 S-MAC 协议。采用其他消息长度和通信模式的实验结果是：在最差情形下，T-MAC 协议的最大吞吐量比 S-MAC 协议低约 70%。不必担心 T-MAC 协议最大吞吐量的下降，这是因为只有在极端载荷条件下

才会发生 T-MAC 协议最大吞吐量下降的问题，而极端载荷是传感器应用（比如消息累积）很容易避免的。

4. 提前休眠问题

从实验中看到：FRTS 机制将最大吞吐量提高约 75%（0.08 条消息每秒对 0.14 条消息每秒， $((0.14-0.08)/0.08) \times 100\% = 75\%$ ），代价是能耗稍有增加。采用全缓存优先机制后，吞吐量又提高约 30%（0.14 条消息每秒对 0.18 条消息每秒， $((0.18-0.14)/0.14) \times 100\% = 28.57\%$ ），但是却没有增加能耗。实验结果如图 2-4（c）所示。

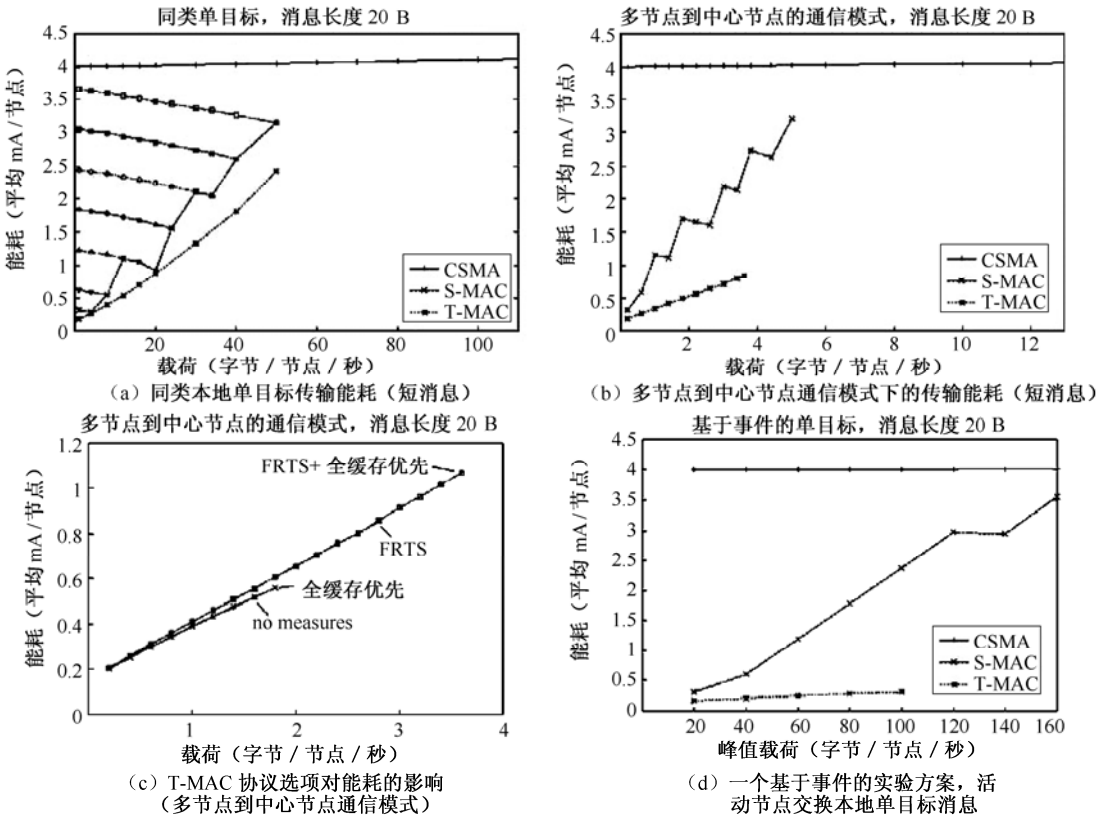


图 2-4 T-MAC 协议仿真实验结果

5. 基于事件的本地单目标传输能耗

现在继续朝一个比较真实的方案进行实验。在这个实验中，网络中每 10 s 发生一个事件。事件平均持续时间 5 s，影响约 9 个节点的一个区域。然后这些节点给其相邻节点发送本地单目标消息，以便通知事件持续时间。相邻节点接收到其中一条消息后以概率 20% 应答。做多次测试，每次采用不同的事件产生速率。实验结果如图 2-4（d）所示。对于 T-MAC 协议，采用旁听回避机制，但是不采用 FRTS 机制和全缓存优先机制。

图 2-4（f）表明 T-MAC 协议的能耗比 S-MAC 协议低得多，特别是事件产生速率增大的时候，T-MAC 协议的能耗优势比 S-MAC 协议更加明显。但是，T-MAC 协议能够处理的最大事件产生速率低于 S-MAC 协议，正像在多节点到中心节点通信模式下的实验结果一样。由

于边沿节点相对较多，所以 T-MAC 协议遇到提前休眠问题。

2.3 伯克利媒介访问控制协议 (B-MAC)

加州大学伯克利分校(UCB)开发的伯克利媒介访问控制(Berkeley Media Access Control, B-MAC)协议用于低功率 WSN。B-MAC 有一个小内核，分解高层功能，其设计和实现均简单。分解一些高层功能和接受高层服务控制使得 B-MAC 协议能够支持许多 WSN 流量。这种最低必须限度 MAC 协议设计模式不同于标准的、完整而庞大的、为一组通用流量而优化的 MAC 协议。

下面描述一种 WSN 应用操作的分析框架，建立监视应用的分析模型。可以使用该模型计算和设置 B-MAC 协议参数，优化应用的总能耗。运用该模型举例说明不同应用变量的效果，包括占空因数、网络密度、采样速率。说明网络服务如何使用 B-MAC 接口来适应当前需求。

2.3.1 B-MAC协议的设计与实现

尽管 B-MAC 是根据监视应用而开发的，但是 B-MAC 灵活性强，允许高效实现其他服务和应用。这些服务包括（但是不限于这些服务）目标跟踪、定位、触发事件报告、多跳路由。

传统 MAC 协议为了获取一组流量的良好性能而进行信道访问仲裁和进行调整。S-MAC 就是为 WSN 设计的一个 MAC 例子。S-MAC 采用传统方法，提供 RTS-CTS 机制进行信道仲裁、隐含终端回避、与相邻节点进行低功率同步操作以及高效大数据量传输的消息分片。S-MAC 不仅是一个链路层协议，而且也是网络与组织协议。随着节点和网络状态的变化，应用和服务必须依靠 S-MAC 的内部策略来调整其操作。但是，节点和网络状态的变化对应用不是透明的。对比之下，B-MAC 协议包含一个媒介访问功能的微小内核，采用干净信道评估 (Clear Channel Assessment, CCA) 和分组退避进行信道仲裁，采用链路层应答保证可靠性，采用低功率侦听 (Low Power Listening, LPL) 进行低功率通信。B-MAC 只是一个链路层协议，具有组织、同步以及在其实现之上进行路由选择之类的网络服务。尽管 B-MAC 协议既不提供隐含终端支持之类的多分组机制、也没有消息分片功能、也不会强制执行某种特定低功率策略，但是 B-MAC 协议有一组接口，该组接口除了提供标准消息接口之外、还允许服务调整其操作。TinyOS 中的消息传输标准接口是：消息发送为 BareSendMsg，消息接收为 ReceiveMsg，时戳和帧分隔符起始 (Start of Frame Delimiter, SFD) 信息为 RadioCoordinator。这些接口允许网络服务调整 B-MAC 机制，包括 CCA、应答、退避以及 LPL。通过控制一组可配置机制，构建在 B-MAC 协议之上的协议就可以就地做出策略决策来优化功耗、时延、吞吐量、公平性或者可靠性。

为了进行有效的碰撞回避，MAC 协议必须能够准确确定信道是否干净，这就称为 CCA。由于周围噪声变化依赖环境，所以 B-MAC 采用软件自动增益控制来估计噪声平面。当假设信道为空闲时，比如正好在发送完一个分组之后或者在无线栈的数据路径没有接收到有效数据之时，不时提取信号强度样值。然后将样值写入 FIFO 队列。利用衰减因子 α 对 FIFO 队列长度中间值进行指数加权移动平均值。FIFO 队列长度中间值作为简单低通滤波器，用于提高噪声平面估计的强壮性。对于典型的无线信道， $\alpha = 0.06$ 、FIFO 队列长度等于 10 时得到最佳

结果。一旦确定了对噪声平面的恰当估计,那么发送分组请求就立即启动无线信道接收信号的监视过程。在各种协议(包括 IEEE 802.15.4)中普遍使用的一种方法是取一个样本、并将其与噪声平面比较。这种门限法得到的结果包含大量虚假正确结果,降低了有效带宽的利用率。由于噪声的信道能量变化大,而分组接收却是相当恒定的信道能量[见图 2-5 (a)],所以 B-MAC 搜索接收信号之外的局外信息,以便使信道能量远在噪声平面之下。假如在信道采样期间找到局外信息,那么 B-MAC 声称信道是干净的,这是因为一个有效分组是绝不可能有远处于噪声平面下方的局外信息。假如采取了 5 个样本而仍然没有找到局外信息,则信道忙。图 2-5 (b) 给出了局外信息检测的功效与门限法的 CC1000 收发信机信号对比。

最基本的机制是允许服务采用 B-MAC 协议提供的 MacControl 接口打开或者关闭 CCA。通过关闭 CCA 就可以在 B-MAC 上面实现时间安排协议。若打开 CCA,则 B-MAC 在发送分组之时采用初始信道退避。B-MAC 不设置退避定时器,而是将事件发送给服务,服务通过 MacBackoff 接口发送该分组。服务或者可能返回初始退避时间或者可能不予理睬事件。假如不理睬事件,那么采用短小随机退避。初始退避完成之后,执行 CCA 局外信息算法。假如信道不干净,那么给服务发送一个事件,以便进行拥塞退避定时。假如没有给定退避时间,那么可采用短小随机退避。打开或者关闭 CCA,以及配置退避时间允许服务改变公平性和有效吞吐量。

B-MAC 提供可选的链路层应答支持。假如采用链路层应答,那么 B-MAC 在接收到一个单目标分组后立即发送应答码。假如发送节点接收到应答,则将发送节点的发送消息缓存器中的应答比特位置位。

B-MAC 通过周期性采样设置电台的占空因数,将周期性采样称为低功率侦听(Low Power Listening, LPL)。LPL 技术类似于 Aloha 中的前导采样,但是适应不同的电台特性。每当节点苏醒时,打开其电台,检查信道活动情况。假如检查到信道活动情况,节点加电开机,保持一段时间的苏醒,这段时间的长度等于接收输入分组所需要的时间。节点接收到分组后返回到休眠状态。假如没有接收到分组(虚假正确),那么超时迫使节点返回到休眠状态。精确信道估计(CCA)对于采用这种方法实现低功率操作非常关键。采用 B-MAC 的噪声平面估计,不仅是为了寻找干净的传输信道,而且还是为了确定信道在 LPL 期间是否处于活动状态。CCA 算法中的虚假正确问题(正如门限算法产生的虚假正确问题一样)导致空闲侦听增多,因而严重影响 LPL 的占空因数。

为了可靠接收数据,前导长度与信道活动情况检查间隔匹配。假如按照周期 100 ms 检查信道,那么前导长度必须至少等于 100 ms,节点才能够苏醒,然后检查信道活动情况,接收前导,最后接收消息。当节点苏醒、采样信道而没有检查到信道活动情况时,则会发生空闲侦听。LPL 样本间隔最大,则信道采样花费的时间最少。发送方式对应前导长度,侦听方式对应检查间隔。提供 8 种方式(对应于 10 ms、20 ms、50 ms、100 ms、200 ms、400 ms、800 ms、1 600 ms 的检查间隔)供选择。协议也可以通过接口设置自己的前导长度和检查间隔。

在 Mica2 Mote 传感器上信道采样的功耗曲线如图 2-5 (b) 所示。图 2-5 (b) 的过程实质上适用于 WSN 的任何 MAC 协议:节点首先从休眠状态开始(a),然后在定时器中断时苏醒(b)。节点初始化电台配置,开始电台的启动过程。电台启动过程(c)等待电台晶体振荡器稳定。稳定后,电台开始进入接收方式(d)。经过接收方式切换时间后,电台进入接收方式(e),可以开始对接收信号能量采样。ADC 开始采集后,关掉电台,分析 ADC 样值(f)。采用 LPL,若是没有检查到信道上的活动情况,则节点返回到休眠状态(g)。结果,所有协

议打开电台的能耗开销相同。不同协议之间的差异在于电台启动之后保持的时间长度以及电台启动的次数。

在 WSN 中，通常每个节点执行单个应用。由于传感器节点内的可用 RAM 和 ROM 极其有限，所以保持 MAC 协议实现代码少非常重要。降低协议复杂性就是减少状态以及降低紊乱情况的可能性。在 TinyOS 上实现 B-MAC，评估 B-MAC 在满足设计目标上的功效。由于 B-MAC 没有 S-MAC 的 RTS-CTS 机制和同步要求，所以 B-MAC 的实现较简单、代码较小，见表 2-2。B-MAC 不妨碍网络协议的高效实现。在 B-MAC 上面实现了相当于 S-MAC 的 RTS-CTS 机制和采用 B-MAC 控制接口的消息分片服务。

表 2-2 B-MAC 和 S-MAC 的字节大小比较（两个协议均在 TinyOS 上实现）

协 议	ROM（字节）	RAM（字节）
B-MAC	3 046	166
含 ACK 的 B-MAC	3 340	168
含 LPL 的 B-MAC	4 092	170
含 LPL 和 ACK 的 B-MAC	4 386	172
含 LPL 和 ACK 的 B-MAC + RTS-CTS	4 616	277
S-MAC	6 274	516

2.3.2 寿命建模

为了计算节点占空因数和寿命，研究周期性感知应用（比如栖息地监视应用所述），将传感器数据传递给一个中心节点。表 2-3 列出了低功率监视应用执行的原始操作、以及采用 CC1000 收发信机时所观测到的开销。这些操作描述了 WSN 的一类代表性电台。Chipcon、Infineon、Motorola 公司生产具有类似特性的电台。B-MAC 协议的后续描述全部采用表 2-3 中的符号和取值。

节点寿命由其总能耗决定。假如寿命最长，那么能耗必须最小。总能量 E 定义毫焦耳/秒或者毫瓦。总能量 E 乘以节点寿命 t_1 得到总能耗。对于 WSN 应用，一个节点使用的能量 E 包括接收 (E_{rx})、发送 (E_{tx})、无线信道上的消息侦听 (E_{listen})、数据采样 (E_d)、休眠 (E_{sleep}) 的能耗，即

$$E = E_{rx} + E_{tx} + E_{listen} + E_d + E_{sleep} \tag{2-15}$$

传感器是 WSN 的一个完整组成部分，在计算节点寿命时是必须考虑的。对传感器采样常常是高开销的，影响节点的寿命。基于 Mainwaring 等人开发的栖息地监视应用的采样参数如表 2-3 所示。在这些应用中，每个节点需要 1 100 ms 用于启动传感器、采样以及数据收集。数据采样周期为 5 min，即应用分组产生速率 $r=1/(5\times60)$ 。有关数据采样的能耗 E_d 为

$$\begin{aligned} t_d &= t_{data}\times r \\ E_d &= t_d\times c_{data}\times V \end{aligned} \tag{2-16}$$

发送能耗 E_{tx} 等于数据分组长度与前导长度之和乘以应用分组产生速率，即

$$\begin{aligned} t_{tx} &= r\times(L_{preamble}+L_{packet})\times t_{txb} \\ E_{tx} &= t_{tx}\times c_{txb}\times V \end{aligned} \tag{2-17}$$

对于均匀采样速率的周期性应用，节点的 n 个相邻节点中的一个节点发送一个分组的时候，该节点检测和接收这个相邻节点发送的分组，而不管该分组的目的地。把一个节点周围

的相邻节点密度称做该节点的邻域大小（Neighborhood Size）。接收相邻节点的数据尽管会缩短接收寿命，但是却允许服务旁听信道和根据信道活动情况做出决策。

表 2-3 一个采用 Mica2 Mote 传感器和 CC1000 收发信机的监视应用的时间和电流消耗
（每种操作的标识符对应图 2-5（b）所示的获取电台样本的活动）

操 作	时间/s		电流/mA	
电台初始化（b）	350E-6	$t_{r_{init}}$	6	$C_{r_{init}}$
打开电台（c）	1.5E-3	$t_{r_{on}}$	1	$C_{r_{on}}$
切换到 RX/TX 方式（d）	250E-6	$t_{rx/tx}$	15	$C_{rx/tx}$
电台采样时间（e）	350E-6	t_{sr}	15	C_{sr}
电台样本评估（f）	100E-6	t_{ev}	6	C_{ev}
接收 1 字节	416E-6	t_{rxb}	15	C_{rxb}
发送 1 字节	416E-6	t_{txb}	20	C_{txb}
采样传感器	1.1	t_{data}	20	C_{data}

可以限制数据接收花费的总时间，计算出数据接收的能耗上限值 E_{rx} ，即

$$\begin{aligned} t_{rx} &\leq n \times r \times (L_{preamble} + L_{packet}) \times t_{rxb} \\ E_{rx} &= t_{rx} \times C_{rxb} \times V \end{aligned} \tag{2-18}$$

这是根据单蜂窝来进行分析的。为了分析多跳应用，需要考虑通过每个节点的路由流量，这是由于每个节点的子节点和相邻节点的子节点。通过某个特定节点的流量不是 r 个分组/秒，而是必须包括该节点和其相邻节点发送的所有分组。函数 $children(i)$ 由多跳路由协议定义。

$$r \times \sum_{i=0}^n (children(i) + 1)$$

到现在为止，寿命模型还是独立于所使用的 MAC 协议。MAC 协议负责空闲侦听时间 t_{sleep} 最小化。在 B-MAC 协议中，只要 B-MAC 采样信道活动情况而又没有活动，则会发生空闲侦听。

为了可靠接收分组，LPL 检查间隔 t_i 必须小于前导时间。因此，得到如下限制条件

$$L_{preamble} \geq [t_i / t_{rxb}]$$

给定检查间隔 t_i 和有关前导长度，就能够计算信道采样的耗时。从图 2-5（b）中可以看到，单个 LPL 电台样本的功耗是 17.3 μJ 。信道侦听的总能耗等于单个信道样本能量与信道采样频率之乘积，即

$$\begin{aligned} E_{sample} &= 17.3 \mu J \\ t_{listen} &= (t_{r_{init}} + t_{r_{on}} + t_{rx/tx} + t_{sr}) \times (1/t_i) \\ E_{listen} &\leq E_{sample} / t_i \end{aligned} \tag{2-19}$$

表 2-4 运行 B-MAC 的监视应用的参数（每个参数都影响节点的总能耗 E ）

符 号	参 数	默 认 值	说 明
C_{sleep}	休眠电流/mA	0.030	针对 Mica2 Mote 传感器
C_{batt}	电池容量/mAh	2 500	
V	电压/mV	3	
$L_{preamble}$	前导长度/（字节）	271	在 Mica2 Mote 传感器上运行的 B-MAC 默认参数值

符 号	参 数	默 认 值	说 明
L_{packet}	分组长度/（字节）	36	影响 B-MAC 性能的应用语义
t_i	电台采样周期/s	100E-3	
n	邻域大小/（节点数量）	10	
r	采样速率/（分组数量/秒）	1/300	
t_1	期望寿命/s	—	

最后，节点必须在剩余的时间内休眠。休眠时间 t_{sleep} 等于每秒内未被其他操作消耗的剩余时间。

$$t_{\text{sleep}} = 1 - t_{\text{rx}} - t_{\text{tx}} - t_{\text{d}} - t_{\text{listen}}$$
$$E_{\text{sleep}} = t_{\text{sleep}} \times C_{\text{sleep}} \times V$$

(2-20)

节点的寿命 t_1 依赖总能耗 E 、电池容量 C_{batt} 。必须使用有效电池容量限制节点寿命。

$$t_1 = [(C_{\text{batt}} \times V) / E] \times 60 \times 60$$

(2-21)

解系统式（2-15）～式（2-20），并将其代入表 2-4 中的参数，就能够得到一个给定网络配置的最低能量。就可以在编译时估计寿命，或者通过实时运行时的离散样值集来计算寿命，离散样值集给网络服务提供重新配置反馈信息。

2.3.3 参数

在典型的 WSN 应用中，通过确定节点的物理位置（影响每个节点的邻域大小 n ）以及合适的采样速率 r 就能够计算出 B-MAC 能够实现的最佳寿命参数。

假如采样速率不变、但是改变网络密度 n ，那么也能够估计相邻节点对节点寿命的影响。采样速率为每 5 min 一个分组[见图 2-5（c）]，解系统式（2-15）～式（2-20）。例如，对于相邻区域 20 个节点的最佳检查间隔是 50 ms，但是对于相邻区域 5 个节点的最佳检查间隔是 100 ms。相邻区域大小影响通过每个节点的流量大小。B-MAC 平衡空闲侦听，以便减少发送时间和接收时间。

若假定一个网络中的每个节点约有 10 个相邻节点，那么最佳 LPL 检查间隔时间按照相同速率变化。提高采样速率，则网络中流量增大（就像增大相邻区域导致周期性应用流量增大一样）。结果，每个节点旁听到更多的分组。必须找出最佳检查间隔时间 t_i ，这样才能够使寿命 t_1 最大。缩短最佳检查间隔时间 t_i ，则导致前导长度变小，一个分组的发送时间和接收时间均变短，对电台采样更加频繁。

平衡较频繁地检查电台，换取分组发送时间变短，如图 2-5（d）所示。付出的代价是空闲侦听比流量模式所要求的多，图 2-5（d）中最大寿命点的左边比发送大于所必需长度的分组的代价严重得多。

2.3.4 自适应控制

寿命模型指出：根据网络条件的变化改变 MAC 协议的参数是有利的。由于 WSN 由低功率不稳定节点组成，所以很可能随着时间的推移消失原来的链路和出现新的链路。节点可以加入和退出网络，相邻区域大小由于物理环境的变化而变化。MAC 协议必须能够适应这

些变化，优化其功耗、时延、吞吐量，以便支持依赖 MAC 协议的服务。分析模型允许节点重新计算检查间隔时间和前导长度。为了解决重新配置问题，建立一组双向接口，允许服务根据当前工作状态通过这些接口来改变 MAC 协议。

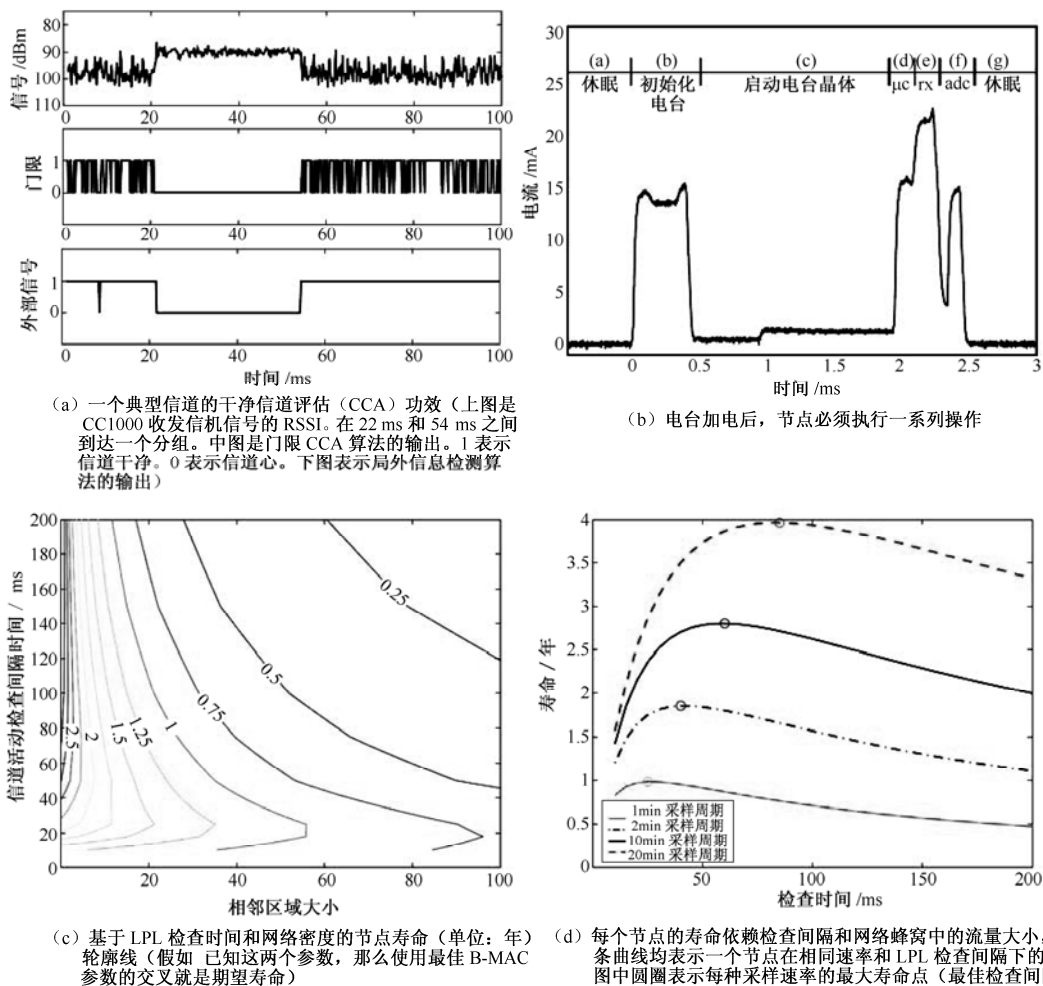


图 2-5 B-MAC 协议描述示意图

通过分解传统 MAC 协议较复杂的组成部分，服务就能够决定在哪些情形下可以使用额外控制。例如，可以选择对每个分组进行 B-MAC 的链路层应答。当应答失败时，服务可以选择重传分组、改变分组的地址或者重新配置 LPL 参数。

在 B-MAC 之上的一种实现方案是 RTS-CTS 信道捕获协议。每当发送一个分组时，首先发送一个 RTS 分组。假如目的节点空闲、并且不会由于其他发送而延迟，那么目的节点回送 CTS 分组。RTS 分组和 CTS 分组均采用 LPL 发送。一旦获取信道后，关闭 CCA 和 LPL，并立即发送数据分组和应答分组。接收到应答后，发送节点和接收节点重新打开 LPL 和 CCA，然后返回到休眠状态。

参 考 文 献

- [1] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification. IEEE Std. 802.11-1999 edition.
- [2] A.Woo and D. Culler. A transmission control scheme for media access in sensor networks. in Proc. ACM/IEEE Int. Conf. Mobile Computing and Networking, Rome, Italy, July 2001, pp.221–235.
- [3] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. in Proc. IEEE INFOCOM, New York, NY, June 2002, pp.1567–1576.
- [4] Wei Ye,J.Heidemann and Deborah Estrin. Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks. IEEE/ACM TRANSACTIONS ON NETWORKING,VOL.12,NO.3,pp.493-506,June 2004.
- [5] W. Ye, J. Heidemann, and D. Estrin. A flexible and reliable radio communication stack on Motes. USC Information Sciences Inst., Tech. Rep. ISI-TR-565, Sept. 2002.
- [6] T. S. Rappaport. Wireless Communications, Principles and Practice. Englewood Cliffs, NJ: Prentice-Hall, 1996.
- [7] M. Stemm and R. H. Katz. Measuring and reducing energy consumption of network interfaces in hand-held devices. IEICE Transactions on Communications, E80-B(8):1125- 1131, 1997.
- [8] A. Varga. The OMNeT++ discrete event simulation system. In European Simulation Multiconference (ESM'2001), Prague, Czech Republic, June 2001.
- [9] Tijs van Dam and Koen Langendoen. An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks. SenSys'03,pp.171-180, November 2003.
- [10] <http://eyes.eu.org/sensnet.htm>.
- [11] <http://webs.cs.berkeley.edu/tos/>.
- [12] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao. Habitat monitoring: Application driver for wireless communications technology. In 2001 ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean, Apr. 2001.
- [13] A. El-Hoiydi. Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks. In Proceedings of IEEE International Conference on Communications, Apr. 2002.
- [14] A. El-Hoiyi, J.-D. Decotignie, and J. Hernandez. Low power MAC protocols for infrastructure wireless sensor networks. In Proceedings of the Fifth European Wireless Conference, Feb. 2004.
- [15] J. Hill and D. Culler. Mica: a wireless platform for deeply embedded networks. IEEE Micro, 22(6):12–24, November/December 2002.
- [16] J.Polastre, J.Hill and D. Culler. Versatile Low Power Media Access for Wireless Sensor Networks. SenSys'04,pp.95-107, November 2004.

- [17] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler. TinyOS: An operating system for wireless sensor networks. In *Ambient Intelligence*. Springer-Verlag, 2004.
- [18] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pages 88–97. ACM Press, Sept. 2002.
- [19] University of California, Berkeley. Mica2 schematics. http://webs.cs.berkeley.edu/tos/hardware/design/ORCAD_FILES/MICA2/6310-0306-01ACLEAN.pdf, Mar.2003.
- [20] University of California, Berkeley. TinyOS CVS Repository at SourceForge. <http://sf.net/projects/tinyos/>, 2004.

第3章 无线传感器网络分配类MAC协议

3.1 流量自适应媒介访问协议 (TRAMA)

美国加利福尼亚大学圣克鲁斯分校研究的流量自适应媒介访问协议 (Traffic-Adaptive Medium Access Protocol, TRAMA) 用做 WSN 的能量高效无碰撞信道访问协议。TRAMA 将不在发送或者不在接收中的传感器节点切换到低功率、空闲状态, 确保单目标、多目标、广播传输没有碰撞, 从而降低能耗。TRAMA 假定将时间分成时隙, 根据每个节点的流量信息, 采用分布式选择法决定时隙的发送节点。对于没有信息发送的节点, TRAMA 不对其分配时隙, 并且允许节点运用流量信息决定自己变成空闲状态、不能侦听信道的时间。TRAMA 是公平且正确的, 不会安排空闲节点作为接收节点, 接收节点不会遇到碰撞问题。

3.1.1 TRAMA协议概述

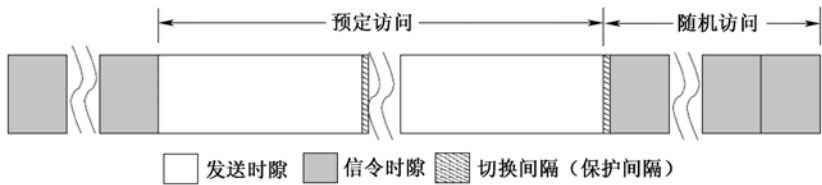
任何竞争类 MAC 协议的控制分组或者数据分组的碰撞概率都随着承载载荷的增加而加重, 从而降低了信道利用率, 进一步缩短了电池的寿命。因此, 需要静态或者动态建立传输时间安排, 允许节点无碰撞地接收数据分组。在无线网络中建立的传输时间安排可以与拓扑无关, 也可以与拓扑有关。开发和最终布置大规模 WSN 所面临的一个主要挑战是各个节点之间传输时间的安排问题, 要求这种安排: ①自适应流量、节点状态、连接的变化; ②延长每个节点的电池寿命。

TRAMA 提供能量高效无碰撞的 WSN 信道访问。TRAMA 的信道访问是能量高效的, 同时又能维持良好的吞吐量, 可容忍时延, 具有公平性。TRAMA 通过以下方法实现能量效率: ①避免在接收节点出现数据分组碰撞的传输时间安排; ②将非预定接收节点的节点切换到低功率工作方式。采用发送节点选举算法实现足够高的吞吐量和公平性, 该算法本身就是公平的, 并且将任意给定源节点或者接收节点周围的竞争数据流作为函数, 从而提高信道复用。TRAMA 根据一跳和两跳范围内节点的身份、当前时隙、流量信息 (说明哪个节点打算给其他节点发送) 推导无碰撞传输时间安排。因此, 一个节点的“休眠时间安排”就是通过该节点和该节点相邻节点的流量的直接函数, 并且在节点交换其有关身份和流量时自动同步。

TRAMA 支持单目标、广播、多目标传输 (即给一个相邻节点集发送)。TRAMA 在两个基本方面不同于 S-MAC 协议 (提供直接的节能机制): ①TRAMA 的媒介访问控制是分配类的, 因此 TRAMA 本身就是无碰撞的, 这不同于属于竞争类的 S-MAC 协议; ②TRAMA 根据当前流量模式, 采用自适应动态方法将节点切换到低功率工作方式, 而 S-MAC 协议是基于预先定义的占空因数的静态法。

TRAMA 类似于节点激活多址访问 (Node Activation Multiple Access, NAMA) 协议。NAMA 使用分布式选举算法实现无碰撞传输。对于每个时隙, NAMA 在每个两跳相邻区域

内只选择一个发送节点,因此发送节点一跳相邻区域内的所有节点都能够无碰撞地接收数据。但是, NAMA 没有解决节能问题。NAMA 采用两跳相邻区域内的节点身份 (ID) 来做出给定节点在特定时隙上的无碰撞信道访问。NAMA 没有解决能量效率问题, 将非发送节点切换到接收工作方式。TRAMA 解决了能量效率问题, 其方法是将一个特定时隙中既没有选做发送节点也不是预定接收节点的节点切换到休眠工作方式。此外, TRAMA 做出传输时间安排时考虑了流量信息, 从而使得 TRAMA 对 WSN 应用更加灵活自如。例如, 事件跟踪应用很可能只在产生一个事件之时产生数据。又如, 监视应用可能连续产生数据。对这两种应用的任何一种, TRAMA 都能够适应, 对其传输时间安排作相应调整, 提供足够的性能和能量效率。



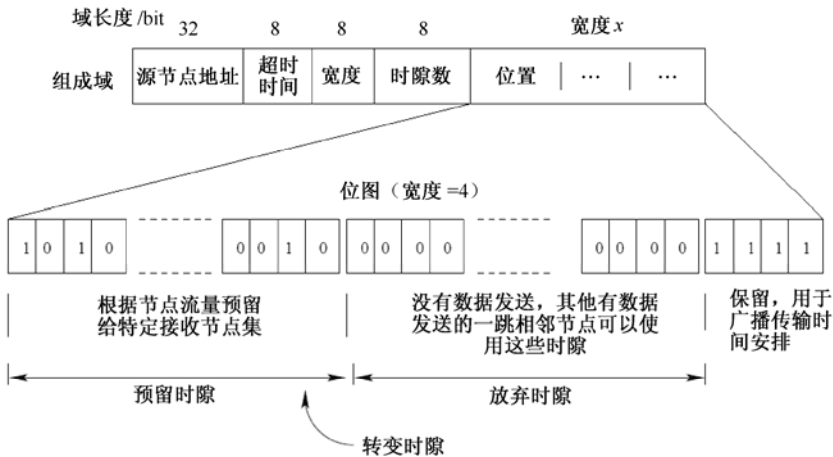
(a) TRAMA 时隙结构

类型	源节点地址	目的节点地址	删除节点数据	增中节点数据	所删除节点的 ID	所增加节点的 ID
----	-------	--------	--------	--------	-----------	-----------

(b) TRAMA 信令头结构

类型	源节点地址	目的节点地址	超时时间	时隙数	位图
					短预定概要

(c) TRAMA 数据头结构



(d) TRAMA 传输时间安排分组格式

图 3-1 TRAMA 结构

3.1.2 TRAMA协议组成

TRAMA 采用流量自适应分布式选择法, 后者根据发送节点广播的时间安排选择接收节点。采用 TRAMA 协议的节点相互交换其两跳相邻区域信息和传输时间安排, 然后选择每个

时隙上的发送节点和接收节点，传输时间安排按照时间顺序指定其传输流的预定接收节点。TRAMA 由三个部分组成：相邻节点协议（Neighbor Protocol, NP）、传输时间安排交换协议（Schedule Exchange Protocol, SEP）和自适应选举算法（Adaptive Election Algorithm, AEA）。NP 和 SEP 用于节点相互交换两跳相邻节点信息及其传输时间安排；AEA 运用相邻区域信息和传输时间安排信息来选择当前时隙的发送节点和接收节点，同时将所有其他节点切换到低功率方式。

TRAMA 假定在时隙化单信道上进行数据和信令传输。图 3-1（a）表示 TRAMA 协议的总体时隙结构。时间分成随机访问段和预定访问段两部分，前者称为信令时隙，后者称为发送时隙。由于 WSN 的数据速率相对较低，所以时隙的长度大于时钟漂移典型值。例如，对于 115.2 kb/s 电台，时隙长度约等于 46 ms，发送 512 B 的应用层数据单元。因此，毫秒级的时钟漂移是可以接受的，而且典型的时钟漂移约为几毫秒。这就可以采用非常简单的时戳机制进行节点同步。假如时钟漂移较小而且又可以使用价格较高的节点，那么节点可以采用 GPS 之类的技术进行时间同步。在下面的 TRAMA 协议描述中，简单假定节点已获得足够精确的时间同步。

NP 协议在随机访问时段使用信令时隙将一跳相邻节点信息发送到其他各个相邻节点，以便所有节点获得一致性的两跳拓扑信息。在随机访问时段，节点进行竞争信道捕获，因此信令分组有可能碰撞。

发送时隙用于无碰撞数据交换和传输时间安排广播。节点使用 SEP 协议与相邻节点交换流量信息或者广播传输时间安排信息。传输时间安排包含来自某个节点的流量的当前信息，即该节点产生的流量的接收节点集。节点在开始真正发送之前必须使用 SEP 广播其传输时间安排。SEP 维护相邻节点之间一致性的传输时间安排信息，周期性更新传输时间安排。

AEA 协议根据从 NP 和 SEP 中获取的信息来选择发送节点和接收节点，从而实现无碰撞传输。因为选择一个特定时隙的发送节点和接收节点对于提高无碰撞传输时间安排的效率是必需的。随机选择发送节点会引起碰撞，选择一个给定时隙的发送节点而不选择其接收节点会导致能量浪费，这是因为所选定发送节点周围的所有相邻节点必须侦听该时隙，即使不接收数据也必须侦听该时隙。选择发送节点而不顾其流量会导致信道利用率下降，这是因为所选定的发送节点可能没有数据需要发送给选定的接收节点。因此，AEA 运用流量信息（即发送节点需要发送给接收节点的流量信息）来提高信道利用率。

根据信道带宽和数据长度，发送时隙的长度固定不变。信令分组通常小于数据分组，因此发送时隙的长度通常是信令时隙的若干倍，这样易于时间同步。TRAMA 研究人员在仿真实现和实验中，将发送时隙设为信令时隙的 7 倍。

3.1.3 访问方式与相邻节点协议

在 WSN 中，节点可能失效（如能量耗尽），也可能增加新节点（如布置新的传感器）。为了包容拓扑的动态性，TRAMA 在随机访问和预定访问之间选择。

TRAMA 按照随机访问方式开始工作，每个节点随机选择一个时隙进行发送。节点只能在随机访问时段入网。随机访问时段与预定访问时段之比（占空因数）跟具体网络有关。在动态性较强的网络中，应该较常使用随机访问。在静态性较强的网络中，随机访问时段之间的间隔可能较大，这是因为只是偶尔才需要包容拓扑变化。在 WSN 中，节点很少移动甚至

不移动, 取决于具体应用类型。因此, 随机访问时段的主要作用是允许节点的加入和删除。可以在随机访问时段进行时间同步。在随机访问时段, 所有节点必须要么处在发送状态, 要么处在接收状态, 这样才能够发送其相邻区更新和接收相邻节点的更新。因此, 随机访问段的长度对能耗起着重要作用。

在随机访问时段, 信令分组可能由于碰撞而丢失, 从而可能导致各个节点的相邻区信息不一致。为了保证一致性及相邻区信息一定的可信度, 应该恰当设置随机访问段长度和信令分组重传次数。研究表明: 对于平均 N 个两跳相邻节点的网络, 信令分组重传次数应该等于 7, 重传间隔等于 $1.44 \times N$, 这样才能够保证 99% 的分组交付率。因此, 随机访问时段长度等于 $7 \times 1.44 \times N$ 。

NP 协议在随机访问期间交换信令分组, 据此收集相邻区域信息。图 3-1 (b) 表示信令分组的头格式。信令分组承载递增式相邻区域更新, 假如没有更新, 则按照连续“继续”信标方式来发送信令分组。每个节点发送递增式更新, 后者包含该节点一跳相邻区域信息, 即由所增加的节点和所删除的节点组成的一个集合。信令分组还用来维护相邻节点之间的连接。一个节点在一段时间内没有接收到某个相邻节点的发送, 则该相邻节点会超时, 重传更新, 以便确保 99% 的成功率。由于节点知道其一跳相邻节点的一跳相邻节点, 所以最终确保整个网络得到一致性两跳相邻区域信息。

3.1.4 传输时间安排交换协议

SEP 协议建立和维护发送节点(即时隙复用)和接收节点(即休眠状态切换)选择所需要的基于流量的传输时间安排信息。节点通过其传输时间安排来获取其数据发送的流量窗口。在预定访问期间将传输时间安排信息周期性广播给一跳相邻节点。

传输时间安排生成的工作原理如下。每个节点根据高层应用的分组产生速率计算 `SCHEDULE_INTERVAL` (传输时间安排的间隔时间)。一个节点的 `SCHEDULE_INTERVAL` 表示该节点可以根据其 MAC 层的当前状态向其相邻节点广播该传输时间安排的时隙数; 然后该节点预先计算间隔 $[t, t + \text{SCHEDULE_INTERVAL}]$ 内的时隙数, 并且对这些时隙赋予最高的优先权[相对于其两跳相邻节点(时隙竞争节点)], 叫做“时隙争夺”。由于这些时隙可能选为该节点的发送时隙, 所以该节点广播这些时隙的预定接收节点。假如一个节点没有足够分组需要发送, 那么该节点就广播放弃相应时隙。其他有数据需要发送的节点就可以使用这些“空闲时隙”。节点在该间隔时间内竞争到的最后一个时隙用于广播其下一个间隔的传输时间安排。例如, 假设节点 u 的 `SCHEDULE_INTERVAL` 为 100 个时隙。在 1 000 个时隙期间, 节点 u 计算其在时间间隔 $[1\ 000, 1\ 100]$ 内竞争到的时隙。假设竞争到的时隙为 1 009, 1 030, 1 033, 1 064, 1 075, 1 098。节点 u 使用其竞争到的最后一个时隙(即 1 098)来广播其下一个间隔 $[1\ 098, 1\ 198]$ 内的传输时间安排。依次持续循环进行。对应竞争到的最后一个时隙的时间固定为该传输时间安排的寿命。

节点使用传输时间安排分组来广播其传输时间安排。由于节点采用 NP 协议获取了两跳拓扑信息, 所以不需要在传输时间安排分组中发送接收节点的地址。节点采用位图传输预定接收节点信息, 位图的长度等于一跳相邻节点数量。位图的每个比特对应一个特定接收节点, 接收节点按照其身份识别码按序排列。这种方案支持的接收节点总数依赖数据时隙长度和已宣布作为接收节点的时隙数量。例如, 假如传输时间安排广播了 16 个时隙, 那么对

于 512/1 024 B 传输时隙，这种方案分别能够支持 256/512 个相邻节点。又如，一个节点有 4 个一跳相邻节点，其身份识别码分别为 14, 7, 5, 4，那么该节点的位图长 4 比特，最高比特对应 14，次高比特对应 7，随后一比特对应 5，最低比特对应 4。采用位图易于支持广播通信和多目标通信。为了广播一个分组，将位图的所有比特置 1，表示所有一跳相邻节点都是该分组的预定接收节点。假如需要将分组按照多目标通信方式传递给相邻节点 14 和 4，那么只需将位图中相邻节点 14 和 4 的对应比特置 1。节点根据其队列中当前流量信息建立赢取时隙的位图。假如队列长度小于传输时间安排中的位图数量，那么所赢取的有些时隙会变得无用。对于这些“空闲时隙”，节点广播一个零位图。两跳相邻区域内的其他节点可以使用零位图时隙。所有赢取时隙变成无用之后的那个时隙称为转变时隙。所有无用时隙都靠近最后一个赢取时隙的前一个时隙，保留最后一个赢取时隙用于广播下一个传输时间安排。这就使得休眠时间达到最大。

图 3-1 (d) 表示传输时间安排分组的格式。其中各组成域的含义如下：

- 源节点地址 (SourceAddr)：表示本传输时间安排分组发送节点的地址；
- 超时时间 (Timeout)：表示本传输时间安排的有效时隙数（从当前时隙开始计算）；
- 宽度 (Width)：表示相邻节点位图的长度（即一跳相邻节点数量）；
- 时隙数 (NumSlots)：表示赢取时隙总数（即本分组包含的位图数量）；
- 预留时隙：表示最后一个赢取时隙，用于广播下一个传输时间安排。

运用每个数据分组发送节点的传输时间安排汇总表。传输时间安排汇总表有助于将传输时间安排分发过程中分组丢失的影响降到最低程度。如图 3-1 (c) 所示，传输时间安排汇总表包括超时时间、时隙数以及当前间隔时间内赢取时隙的位图。位图的长度等于时隙数，用来表示该节点是否在相应时隙发送还是放弃相应时隙。注意：为了不致引起过高的开销（在当前的 TRAMA 设计中，传输时间安排汇总表开销是每个数据分组 6 B），传输时间安排汇总表不携带预定接收节点的信息。这就要求即使在面对传输分组丢失的条件下，也仍然需要维护一跳相邻节点传输时间安排之间的同步（正如在证明 TRAMA 中所指出的，TRAMA 协议的正确性不会受到没有同步的传输时间安排的影响）。例如，接收节点通过检查传输时间安排汇总表中的时隙数和位图就能够更新或者重新同步其存储的传输时间安排信息。每个传输时间安排有一个超时时间，在这个超时时间结束之前不允许节点改变这个传输时间安排。这就要求确保一跳相邻区域内传输时间安排的一致性。

节点维护其所有一跳相邻节点的传输时间安排信息。一个节点只要有最高两跳优先权，就可以查询传输时间安排信息，以便决定自己最后是否真正发送（即该节点有数据需要发送而使用该时隙），或者将该时隙放弃而给相邻区域内其他节点。基于此决策，该节点或者运用从数据分组中获取的传输时间安排汇总表（假如该节点正在接收）中的少量信息更新自己的传输时间安排信息，或者通过假定发送（假如该节点不是发送节点的预定接收节点，并且正处在休眠状态）来更新自己的传输时间安排信息。对于后一种情况，该节点的传输时间安排处在未同步状态，一直持续到该节点根据发送节点随后发送的数据分组中的传输时间安排汇总表来验证或者更新未同步状态时为止。

所有节点在发送节点的转变时隙中侦听信道，以便同步其传输时间安排。例如，假如节点 u 连续在不同时隙给某个特定相邻节点发送，该相邻节点由于其一个竞争节点是节点 u 的隐含节点而不发送分组，那么节点 u 关于相邻节点的传输时间安排就是未同步的。假如节点 u 在转变时隙没有侦听信道，转变时隙是当前传输时间安排间隔的最后一个时隙，相邻节点在

该时隙内发送，那么节点 u 可以假定相邻节点正在发送相应于转变时隙的数据更新其相应传输时间安排。从此刻开始直到相邻节点广播的传输时间安排期满为止的一段时间内，节点 u 认为相邻节点放弃赢取的时隙，从而重新使用该时隙。假如节点 u 试图使用相邻节点赢取的这个时隙（相邻节点实际上在该时隙内发送），就会引起碰撞。因此，相邻节点之间的传输时间安排可能是未同步的，而且这种未同步状态只有一直持续到转变时隙，节点必须在发送节点的转变时隙内侦听信道。

节点在广播传输时间安排的时候，除了有起始竞争到的时隙之外，还有可能获取一些额外的发送时隙。为了避免传输时间安排分组发送之时出现非一致性和碰撞问题，节点应该总是只在先前已广播的超时时间即将结束之时发送传输时间安排分组。由于相邻节点假定转变时隙之后的全部时隙都已放弃时隙，所以传输时间安排可能不同步。因此，在先前已广播的超时时间之前发送传输时间安排分组有可能发生与相邻节点的碰撞。下面将描述如何使用传输时间安排信息来自适应决定节点状态。

3.1.5 自适应选举算法

一个节点在其竞争节点集中具有最高优先级，则选择该节点发送。节点 u 的竞争节点集就是其两跳相邻区域内所有节点组成的集合。节点 u 在时隙 t 的优先权定义为节点 u 的身份和时隙 t 的串联伪随机散列函数，或者表示为

$$\text{prio}(u, t) = \text{hash}(u \oplus t) \tag{3-1}$$

假定节点身份（ID）是唯一的，各个节点是同步的，所有节点在任一给定时隙计算出具有相同优先级。但是，假如选出的节点没有数据发送，那么该时隙就被浪费。此外，节点没有休眠状态，所以自由地给其一跳相邻节点发送。

为了提高能量效率，TRAMA 协议尽可能地将节点切换到休眠状态，并且尽量重复使用那些选出发送节点而没有使用的时隙，以便提高带宽效率。一个已选节点如果没有分组需要发送，则可以放弃其发送时隙，其他节点就可以使用该时隙。节点与其相邻节点交换当前流量信息，以便有效利用低功率空闲电台方式和实现时隙的重复使用。

在已安排的访问周期中的任一给定时隙 t ，根据给定节点 u 的两跳相邻区域信息以及节点 u 的相邻节点广播的传输时间安排确定给定节点 u 的状态。节点的状态包括：发送状态（TX）、接收状态（RX）、休眠状态（SL）。

在任一给定时隙 t ，节点 u 同时满足以下两个条件则处在发送状态：①节点 u 在其竞争节点集中具有最高优先级 $\text{prio}(u, t)$ ；②节点 u 有数据需要发送。

一个节点若是当前发送节点的预定接收节点，则处在接收状态；否则，该节点可以切换到休眠状态，这是因为该节点没有参与任何数据交换。这就意味着：一个节点若不是已选发送节点，则可以通过查询已选发送节点发送的传输时间安排来决定是否处在接收状态；假如发送节点在当前时隙没有数据发送给该节点，那么该节点就可以进入休眠状态。

每个节点执行 AEA 算法，根据当前节点优先级（其两跳相邻区域内）和一跳相邻节点广播的传输时间安排，决定其当前状态（TX、RX、SL）。图 3-2 给出了 AEA 算法的伪码。表 3-1 将给出 AEA 描述采用的一些基本术语和符号。

```

1 计算tx(u), atx(u) and ntx(u)
2 if (u = tx(u)) then
3     if (u:isScheduleAnnouncedF orTx = TRUE) then
4         let u:state = TX
5         let u:receiver = u:reported:rxId
6         Transmit the packet and update the announced schedule
7     else if (u:giveup = TRUE) then
8         call HandleNeedTransmissions
9     endif
10 else if (tx(u) 2 N1(u)) then
11     if (tx(u):announcedScheduleIsV alid = TRUE AND tx(u):announcedGiveup = TRUE) then
12         call HandleNeedTransmissions
13     else if (tx(u):announcedScheduleIsV alid = FALSE OR tx(u):announcedReceiver = u) then
14         let u:mode = RX
15     else
16         let u:mode = SL
17         Update schedule for tx(u)
18     endif
19 else
20     if (atx(u) hidden from tx(u) AND atx(u) 2 PTX(u)) then
21         if (atx(u):announcedScheduleIsV alid = TRUE AND atx(u):announcedGiveup = TRUE) then
22             call HandleNeedTransmissions
23         else if (atx(u):announcedScheduleIsV alid = FALSE OR atx(u):announcedReceiver = u) then
24             let u:mode = RX
25         else
26             let u:mode = SL
27             Update schedule for atx(u)
28         endif
29     else
30         call HandleNeedTransmissions
31 endif
32 procedure HandleNeedTransmissions
33 if (ntx(u) = u) then
34     let u:state = TX
35     let u:receiver = u:reported:rxId
36     Transmit the packet and update the announced schedule
37 else if (ntx(u):announcedScheduleIsV alid = FALSE and ntx(u):announcedReceiver = u) then
38     let u:mode = RX
39 else
40     let u:mode = SL
41     Update the schedule for ntx(u)
42 endif

```

图 3-2 AEA 算法的伪码描述

表 3-1 符号与术语

符 号	术 语
$N2(u)$	节点 u 的两跳相邻节点组成的集合
$N1(u)$	节点 u 的一跳相邻节点组成的集合
$CS(u)$	节点 u 的竞争节点集合是节点 u 的两跳区域内的节点组成的集合，即 $\{u \cup N1(u) \cup N2(u)\}$
$tx(u)$	绝对赢取节点是 $CS(u)$ 中具有最高优先级的节点
$atx(u)$	备用赢取节点是节点 u 的一跳相邻节点中优先级最高的节点，即 $\{u \cup N1(u)\}$
$PTX(u)$	可能发送节点集是集合 $\{u \cup N1(u) - atx(u)\}$ 中满足等式条件②的所有节点组成的集合
$NEED(u)$	必需竞争节点集是集合 $\{PTX(u) \cup u\}$ 中需要额外发送时隙的节点组成的集合
$ntx(u)$	必需发送节点是节点集 $NEED(u)$ 中具有最高优先级，且包含有效同步的传输时间安排的那个节点

节点的状态依赖于绝对赢取节点和其一跳相邻节点广播的传输时间安排。从节点 u 来看，任一时隙 t 的绝对赢取节点可以是：①节点 u 本身；②节点 u 的两跳相邻区域中的节点 v ，若是来自节点 v 的隐含节点，则需要考虑备用赢取节点 $atx(u)$ ；③节点 u 的一跳相邻区域中的节点 w 。

一个节点只要变成一个特定时隙的绝对赢取节点，并且已经广播了该时隙的非零位图，那么就知其两跳相邻区域内没有其他节点在该时隙上发送。因此，该节点就可以给其预定接收节点进行无碰撞的发送。一个节点不是绝对赢取节点的时候，就不能确定具体时隙的真正发送节点。

为了避免绝对赢取节点在没有数据发送之时浪费时隙，TRAMA 协议连续跟踪能够使用额外时隙发送其数据的那些节点。TRAMA 首先计算在当前时隙有可能发送的节点集，将其保存在可能发送节点集之中。可能发送节点集包含一跳相邻区域内能够发送且不会产生碰撞的全部节点。一个节点当且仅当在其两跳相邻区域内具有最高优先级的时候才能够无碰撞地发送。因此，节点使用有效信息检查其一跳相邻区域内的可能发送节点。节点由于无法知道其一跳相邻节点的全部两跳相邻区域信息，所以只能检查该相邻节点在被该相邻节点两跳相邻节点所知道的节点中是否具有最高优先级。换言之，对于节点 u 的一个一跳相邻节点，比如节点 y ，应该在 $PTX(u)$ 中满足以下条件：

$$prio(y) > prio(x) \quad \forall x, x \in N1(N1(y)) \text{ 且 } x \neq N1(y) \tag{3-2}$$

必需竞争节点集是可能发送节点集的子集，只包含那些有数据需要发送的节点。必需竞争节点集还包括节点 u 还没有对其做出传输时间安排（有效安排）的那些节点（因为节点 u 不知道这些节点是否有数据需要发送）。

除非备用赢取节点是绝对赢取节点的隐含节点，并且属于可能发送节点集，否则绝对赢取节点是假定的发送节点。对于前一种情况，备用赢取节点是假定的发送节点。只要假定发送节点放弃发送，则检查必需竞争节点集，选择必需竞争节点集中优先级最高的节点作为必需发送节点 $ntx(u)$ 。假定发送节点的传输时间安排没有列出的节点可以切换到休眠方式，以节省能量。这对于每次有少数几个节点产生数据并且需要将数据发送给少数几个接收节点（组成的集合）的情形非常有利。

3.1.6 TRAMA的性能

通过仿真评估 TRAMA 协议，以及比较 TRAMA 与竞争类协议、分配类协议的性能，竞争类协议采用 CSMA、IEEE 802.11、S-MAC，分配类协议采用 NAMA。

使用 QualNet 作为仿真平台，所有仿真实验使用的物理层模型都是基于 TR1000 的模型。TR1000 是 WSN 使用的典型电台，传输距离短，数据传输速率低（最大 115.2 kb/s），支持内置低功率电台休眠状态，美国加州大学伯克利分校的 Mote 传感器就是采用这种电台。TR1000 发送、接收、休眠方式下的平均功耗分别为 24.75 mW、13.5 mW、15 μ W；工作方式切换最大时间 20 μ s；调制方式 ASK；接收门限为 -75 dBm。在所有实验中，50 个节点均匀分布在 500 m×500 m 的区域中。每个节点的传输距离 100 m，每个节点平均有 6 个一跳相邻节点，两跳相邻区域内平均有 17 个节点。考虑两种不同类型的流量载荷，根据到达间隔时间按指数分布的统计规律生成节点流量。采用这种流量来重点测试协议在不同到达速率下的性能，另外还测试在几种典型 WSN 进行数据采集应用驱动下的 TRAMA 性能。下面将详细描述流量方案和其他仿真参数。

1. 协议参数

在两种流量方案中，将 TRAMA 的协议参数 SCHEDULE_INTERVAL（传输时间安排间

隔) 固定为 100 个发送时隙。信令分组的最大长度固定为 128 B, 其传输时隙长度为 6.82 ms, 包括工作方式切换的保护时间。发送时隙长度是信令时隙长度的 7 倍, 最大支持 896 B 的传输。随机访问时段固定为 72 个时隙, 每隔 10 000 个发送时隙出现一个随机访问时段。

S-MAC 是竞争类信道访问协议, 采用周期性休眠来节省能量。采用 SYNC 分组来建立休眠时间安排, 每隔 SCHEDULE_INTERVAL 就交换 SYNC 分组。占空因数决定休眠间隔的长度。

将 SYNC_INTERVAL 设为 10 s, 改变占空因数 (10% 和 50%)。所有节点时间同步, 因此选择 S-MAC 协议, 允许侦听信道, 休眠周期全网同步。从实验中观察到 S-MAC 协议需要较长时间与其相邻节点建立起侦听/休眠时间安排, 这是因为 S-MAC 协议没有合适的相邻节点寻找协议, 而必须依靠 SYNC 分组。SYNC 分组只发送一次, 而且是不可靠发送。因此, 允许 20 s 的准备时间, 以便稳定相邻节点信息。由于分配类 MAC 协议的排队时延较长, 所以允许在结束仿真前再花一些时间来交付排队缓存分组。仿真时间 400 s, 仿真结果是多次仿真实验结果的平均结果。

2. 综合数据的产生

实验目的是测试 TRAMA 在网络中所有节点根据某种分布规律产生数据流量时的性能。使用指数分布到达间隔时间来产生数据, 传输速率变化范围为 0.5~2.5 s。一个节点每当发送一个分组时就随机选择一个相邻节点作为下一跳节点。分别测试单目标数据产生和广播数据产生。性能参数如下

- 平均分组交付率: 指平均所有节点的所收分组数量与所发送分组数量的百分比。对于广播数据, 一个分组只有被其所有一跳相邻节点所接收到的时候才计算接收到一个分组。
- 休眠时间百分比: 指整个网络的休眠时隙数量与总时隙数的平均百分比。
- 均排队时间: 指分组被交付到接收节点的平均时延。
- 平均休眠间隔时间: 指休眠间隔时间的平均长度。这是对电台工作方式切换次数的测量。工作方式切换涉及短暂功耗, 所以工作方式频繁切换会导致能量浪费。

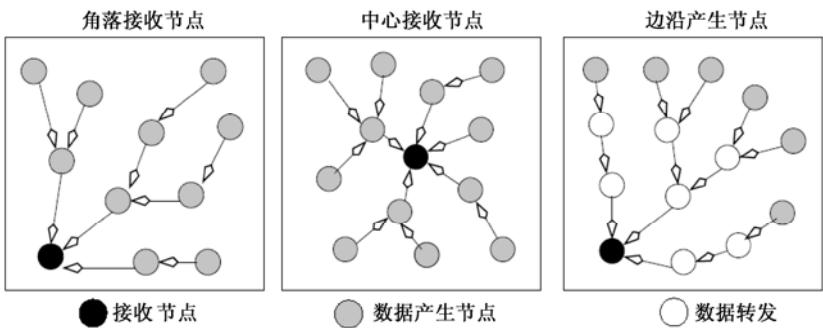


图 3-3 数据采集应用

3. 数据采集应用

假定在实验中一个中心节点收集所有传感器的数据。中心节点发送一个广播查询, 请求所有传感器的数据。传感器接收到广播查询后, 利用自己周期性产生的数据响应。在实验中

实现一个简单的反向路径路由，用于将传感器的数据转发给中心节点。图 3-3 表示用来研究的三个不同的方案。第一种方案，数据收集节点（中心节点）放置在一个角落位置；第二种方案，数据收集节点放置在中心位置。

在前两种方案中，所有传感器均用其周期性产生的数据响应。因为数据累积是为了实现流量最小化，并且是有利的，所以在第三种方案中列举了数据累积。在该方案中，只有边沿节点产生数据流量，并且假定边沿节点进行数据累积以及将其数据附加到父节点上。为了测量性能，对综合数据运行所定义的性能参数进行测量。按照中心节点接收到的数据总数与所有传感器发送的数据总数的百分比来测量平均分组交付率，而不是综合数据流量产生中使用的每跳交付率。

4. 性能仿真结果

综合数据流量的分组交付率、平均排队时延、休眠时间百分比、平均休眠间隔时间长度分别如图 3-4 (a) 至图 3-4 (d) 所示。给出了 S-MAC 协议在两种不同占空因数（即 50% 和 10%）下的实验结果。实验结果指出：一般地，分配类 MAC 协议的分组交付率优于 IEEE 802.11、CSMA、S-MAC，其主要原因在于传输期间始终保证无碰撞传输。当所有节点产生广播数据流时，分配类 MAC 协议的分组交付率优势更加明显。在 IEEE 802.11、CSMA、S-MAC 中，广播是不可靠的，易受隐含节点的碰撞。当增加载荷的时候，隐含节点碰撞导致广播交付率严重下降。对于 IEEE 802.11 和 S-MAC，相对广播数据流，采用 RTS/CTS/DATA/ACK 机制进行的单目标传输交互提高了分组交付率，这是因为通过分组交互以及碰撞回避操作减轻了隐含节点的碰撞。

对比于 TRAMA，10% 占空因数下的 S-MAC 协议表现出较高的休眠时间百分比。但是，平均休眠时间间隔长度却低得多，这是由于电台工作方式切换增加了开销，从而导致总节能的下降。即使假定建立起节点间同步的侦听/休眠传输时间安排，选用 S-MAC 协议，效果仍然如此。TRAMA 的性能不受以离散间隔时间入网节点的影响，这是因为 TRAMA 不要求同步的侦听/休眠传输时间安排。

在下面的实验中，只考虑 10% 占空因数下的 S-MAC 协议，因为其性能优于 50% 占空因数下的 S-MAC 协议。

另一方面，分配类 MAC 协议产生较高的平均排队时延。在测试平均排队时延的时候，考虑已成功交付分组的时延。因此，TRAMA 和 NAMA 交付的分组多于竞争类 MAC 协议，从而减少了高层的重传。因此，从应用层观测到的端到端时延相当于竞争类 MAC 协议的端到端时延。TRAMA 的平均排队时延高于 NAMA，这是因为传播传输时间安排增加了开销。每隔一个 SCHEDULE_INTERVAL，就要使用一个发送时隙来广播传输时间安排，从而降低了数据传输的有效信道访问概率。这种方案不适用于流量自适应选择，这是因为网络中流量均匀分布，所有节点周期性产生数据流量。对于单目标流量和广播流量，TRAMA 的吞吐量与 NAMA 相当，但是比竞争类 MAC 协议高出许多。能量高效协议 S-MAC 的吞吐量与 IEEE 802.11 相当，但是其时延稍高于 IEEE 802.11 和 CSMA，并且随着侦听周期的占空因数的减小而增大，其原因归功于休眠周期的作用。对于 50% 的占空因数，S-MAC 协议的单目标数据传输时延小于 IEEE 802.11，这是因为 S-MAC 协议在 50% 占空因数时在休眠方式和侦听方式之间频繁切换。这等效于节点在大部分时间被唤醒，因而时延较小，这是因为 S-MAC 协议没有竞争解析算法。

TRAMA 协议的节能主要依赖于流量模式，S-MAC 协议的节能依赖于占空因数。总的节能依赖休眠时间百分比和平均休眠间隔时间。休眠时间百分比没有考虑电台工作方式频繁切换引起的分组丢失。平均休眠间隔时间定量描述涉及电台工作方式切换次数。平均休眠间隔时间越长，则越优先选择，这是因为平均休眠间隔时间越长意味着电台工作方式切换越少，因此节能越多。这个结果指出：直观上，广播数据流的休眠时间百分比小于单目标数据流的休眠时间百分比。S-MAC 协议的休眠时间百分比随着占空因数的减小而递增，减小占空因数的代价是时延的增大。若占空因数较小，那么随着数据流量的递增，吞吐量急剧下降，变得越来越陡峭。对于广播分组，50% 占空因数下的吞吐量小于 10% 占空因数下的吞吐量。采用常规载波侦听进行广播更易遇到隐含节点的碰撞。512 B 数据的定时结构采用低占空因数，这样能够减轻信道的竞争，从而减少碰撞。注意：一个广播分组只有被其全部相邻节点接收到的时候才能计算交付分组。

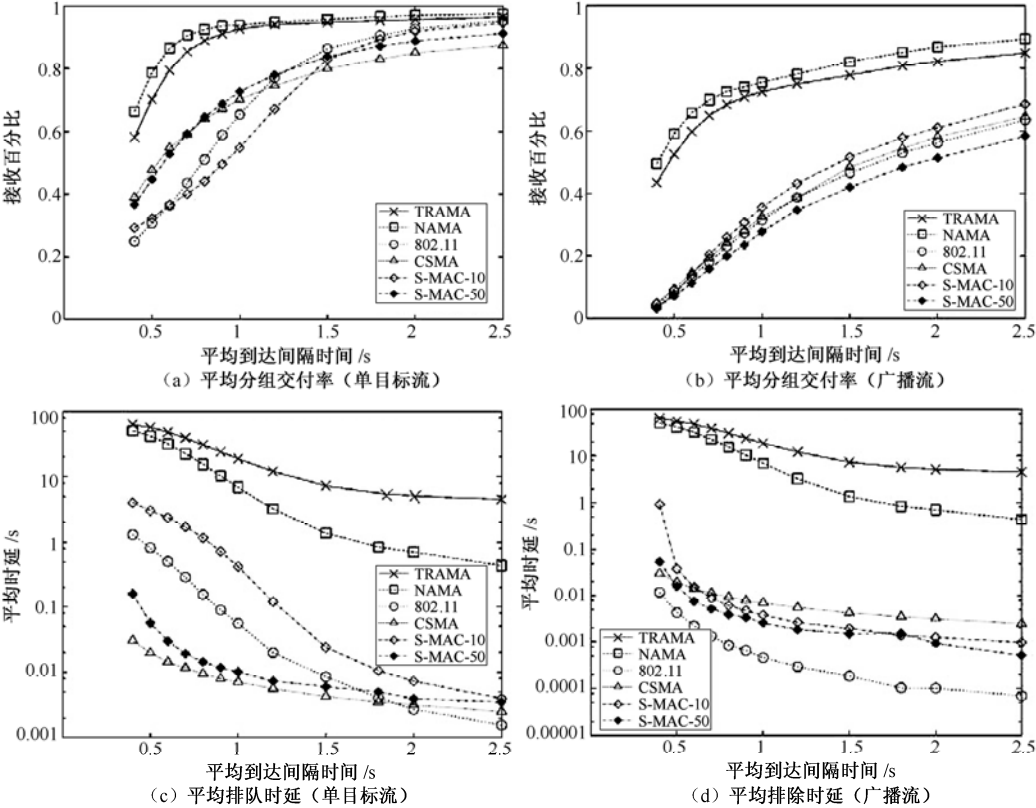


图 3-4 TRAMA 协议的仿真结果

3.2 分布式随机时隙安排协议 (DRAND)

美国北卡罗来纳州州立大学开发的分布式随机时隙安排协议 (Distributed RAND, DRAND) 是无线 Ad Hoc 网络集中式 TDMA 算法 (Randomized Time Slot Scheduling, RTSS) 的分布式、强壮、可扩展的实现，是 RAND 的第一个全分布式版本。DRAND 对两跳距离内

节点进行添色，因此两跳范围内各个节点分得不同的颜色。DRAND 能够对任意图进行时隙分配。DRAND 算法按照循环方式工作，要求每个节点维护其一跳相邻区域内的状态，不要求每个节点同步到每轮的时间边界上，能够高效地自适应本地拓扑变化，不会产生传输时间安排的全网开销。

3.2.1 TDMA时隙分配问题定义

将网络表示为一张图 $G=(V, E)$ ， V 表示网络节点组成的集合， E 表示网络边组成的集合。当且仅当节点 u 和 v 属于集合 V 、且能够相互接收到对方的发送（即所有边都是双向边）时，才存在边 $e=(u, v)$ 。将时间分成非重叠等长度的时间帧，将时间帧分成 MaxSlot 个非重叠等长度的时隙，各个时隙依次标号为 1 到 MaxSlot，MaxSlot 足够大且能够处理任意输入图的全部分配策略。

当且仅当两个节点 u 和 v 同时发送造成在某个节点上发生无线干扰，就说这两个节点 u 和 v 处在冲突中。在广播传输时间安排方式下，两跳相邻区域内所有节点之间都可能发生冲突。在单目标传输时间安排方式下，发送节点和接收节点的一跳相邻区域内所有节点之间都可能发生冲突。

将时隙分配问题正式定义：在给定输入图和冲突定义条件下，给每个节点分配一个时隙，若任意两个节点处在冲突中，则这两个节点不会有相同时隙。通过以下三个性能参数可以确定时隙分配算法的性能。

① 最大时隙号：TDMA 时隙分配问题是图形添色问题的直接延伸问题，后者的目标就是使用最少的颜色给一张图的顶点添颜色，任意两个相邻节点没有相同的颜色。用一条边连接每对冲突节点，然后给这张新图添色就类似于计算 TDMA 传输时间安排，这是因为任意两个冲突节点不能共享同一个 TDMA 时隙。试探法常常报告给定添色算法的所有进程完成颜色分配所需要的最大色数，将这个最大色数叫做最大时隙号，按照图形添色术语也叫做最差情形色数。

② 运行时间： V 中所有节点在所有算法执行中确定其时隙所需要的最大时间。

③ 消息复杂性： V 中所有节点在所有算法执行中确定其时隙而发送的消息最大数量。

3.2.2 DRAND算法详述

在描述 DRAND 算法过程中，假定是广播方式，很容易将下面的 DRAND TDMA 时隙分配算法描述扩充到其他冲突关系中。DRAND 是 RAND 的分布式实现，RAND 是集中式时隙分配算法。因此，DRAND 达到的信道效率与 RAND 相同，但是其平均时间和消息复杂性为 $O(\delta)$ ， δ 表示网络中任意节点的两跳相邻节点的最大个数。

DRAND 循环运行。根据对网络时延的估计动态调整每轮的执行时间。但是，DRAND 算法不要求每个节点同步到每轮的时间边界上。每个节点维护 4 种状态：空闲 (IDLE)、请求 (REQUEST)、同意 (GRANT)、释放 (RELEASE)，图 3-5 表示 DRAND 状态图。

开始，节点 A 处在空闲状态。在空闲状态，A 投硬币，出现正面、反面的概率各占一半。一个节点若是得到硬币正面，则抽彩，抽彩有一定的成功概率。节点若是中彩，则与其相邻节点交换消息，协商选择一个时隙。更准确地说，每个节点 j 维护一个估计 C_j ，即对其一跳

相邻节点和两跳相邻节点中还没有确定其时隙的那些节点的估计。假如自从最近一次抽彩以来所经过的时间大于 T_A ，那么 A 抽彩，将赢取概率设为 $p_A=1/k$ ，其中将 k 设为 C_j 的最大值，对 A 的一跳相邻节点和两跳相邻节点中的所有 j 。

设 j 表示 A 迄今为止抽彩的次数，则称 A 处在第 i 轮中。假如 A 中彩，那么 A 转移到 REQUEST 状态，给其一跳相邻节点广播一条 $request_i$ （请求）消息。假如 A 没有中彩，那么 A 继续保留在 IDLE 状态。设 $T_A=3d_A$ ， d_A 表示 A 对其消息单向传输时延最大值的近似估计（A 只要接收到其他节点对其所发请求的响应，就可以得到这个估计）。

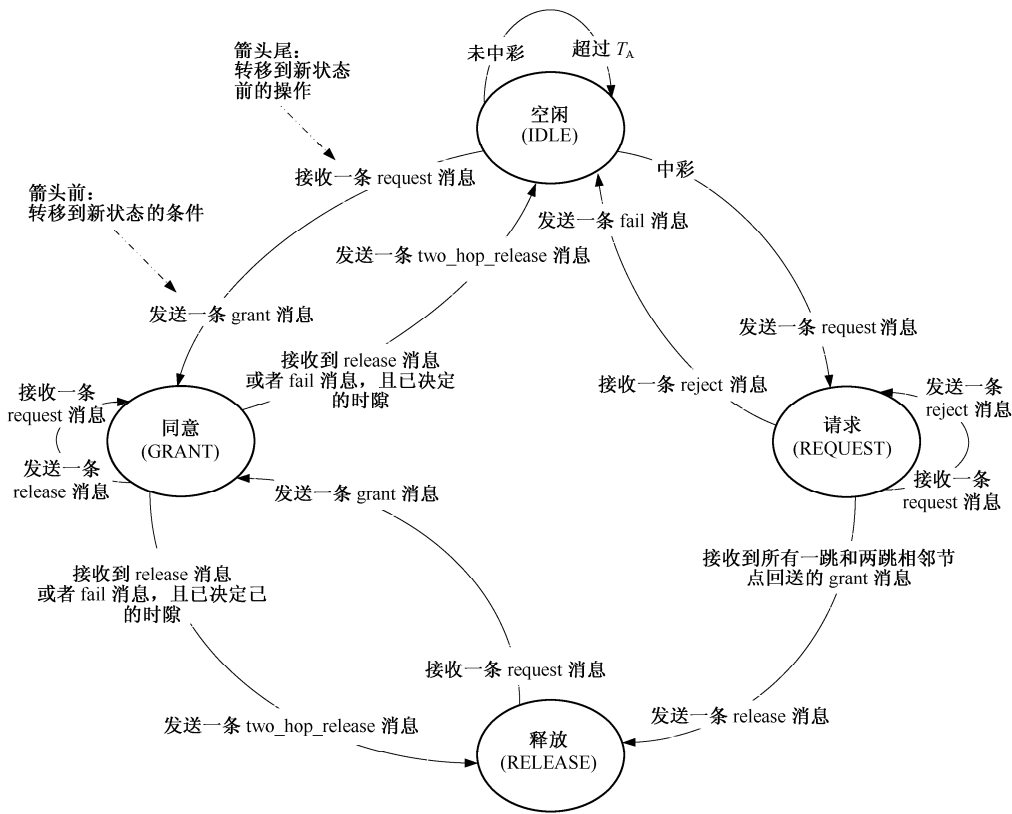


图 3-5 DRAND 状态图（在箭头始端标出进行状态转移的条件，在箭头末端标出转移到新状态前的操作）

假如一个相邻节点 B 接收到 A 发送的 $request_i$ ，并且 B 处在 IDLE 状态或者 RELEASE 状态，那么 B 转移到 GRANT 状态，然后给 A 回送一条 $grant_i$ （同意）消息。B 只有满足如下条件下之一才会处在 IDLE 状态或者 RELEASE 状态：① B 的相邻节点没有发送过 request 消息（注意由于 A 和 B 不同步，所以有可能 B 处在不用于 A 的某个循环中）；② B 迄今为止还没有给任何节点回送过 grant 消息；③ 假如 B 已经给另外一个节点（不是 A）回送过一条 grant 消息，那么 B 已经接收到 fail（失败）消息或者 release（释放）消息。B 在发送 $grant_i$ 时， $grant_i$ 包含 B 一跳相邻节点选定的时隙信息（B 接收到其一跳相邻节点发送的 release 消息后就知道这些时隙）。

假如一个相邻节点 B 接收到 A 发送的 $request_i$ ，并且 B 处在 REQUEST 状态或者 GRANT 状态，那么 B 给 A 回送一条 $reject_i$ （拒绝）消息。A 接收到任何节点回送的 $reject_i$ 消息后，

给其所有一跳相邻节点广播一条 $fail_i$ 消息，然后转移到 IDLE 状态。B 接收到 A 广播的 $fail_i$ 消息后且 A 发送的 $request$ 消息使 B 的当前状态转移到 GRANT 状态，有两种处理情况：①假如 B 还没有确定自己的时隙，那么 B 转移到 IDLE 状态；②假如 B 已经确定了自己的时隙，那么 B 转移到 RELEASE 状态。图 3-6 (a) 给出一个失败循环例子：节点 B 在接收到 A 发送的 $request$ 消息之前已经给其另外一个一跳相邻节点发送了一条 $grant$ 消息。

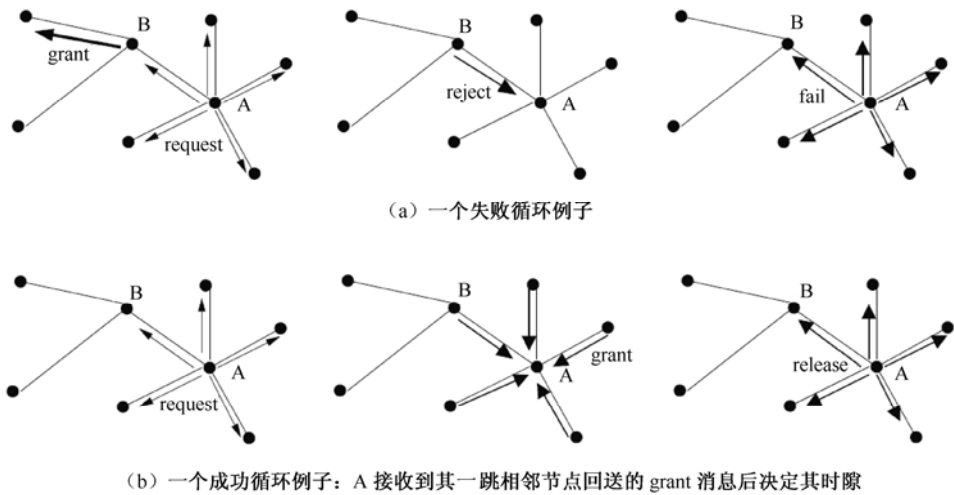


图 3-6 DRAND 应用举例

假如 A 在时间 d_A 内没有接收到一跳相邻节点 B 回送的 $grant_i$ 消息或 $reject_i$ 消息，那么 A 给没有回送 $grant_i$ 消息的所有一跳相邻节点重新发送 $request_i$ 消息。假如一个节点 i 接收到另一个节点 j 发送的 $request_i$ 消息，但是节点 i 已经给节点 j 回送了一条 $reject_i$ 消息，那么节点 i 给节点 j 重新回送一条 $reject_i$ 消息。只要 A 接收到 $grant_i$ 消息或 $reject_i$ 消息，A 就估计消息传输时延约等于 $request_i$ 消息和所收消息 ($grant_i$ 或 $reject_i$) 的时戳之差。若新估计值大于 d_A 的当前值，则将 d_A 设为新估计值。

当 A 接收到其所有一跳相邻节点对其所发 $request_i$ 消息所回送的 $grant_i$ 消息后，A 就决定自己的时隙为本轮之前其两跳相邻节点还未占用的时隙中时隙号最小的那个时隙 ($grant$ 消息中携带时隙信息)。然后，A 转移到 RELEASE 状态，给其一跳相邻节点广播一条 $release_i$ (释放) 消息， $release_i$ 包含 A 所选定的时隙。图 3-6 (b) 给出了一个成功循环的例子。

A 的一跳相邻节点接收到 $release_i$ 消息后，有两种处理情况：该相邻节点若是还没有确定自己的时隙，那么转移到 IDLE 状态；该相邻节点若是已经确定了自己的时隙，那么转移到 RELEASE 状态。然后该相邻节点将该 $release_i$ 消息重新广播给自己的一跳相邻节点。将已转发的 $release_i$ 消息称为 two-hop-release_i (两跳释放) 消息。节点接收到 two-hop-release_i 消息后就能够估计 k 。假如节点 B 给 A 发送完 $grant_i$ 消息后在时间 d_B 内没有接收到 $fail_i$ 消息或者 $release_i$ 消息，那么 B 重发 $grant_i$ 消息。假如一个节点接收到一条 $grant_i$ 消息，这条消息是针对 A 先前发送的 $fail_i$ 消息或者 $release_i$ 消息，那么 A 重新给这个节点发送 $fail_i$ 消息或者 $release_i$ 消息。

3.2.3 DRAND正确性

定理 3.1 执行 DRAND 的结果是得到无碰撞的 TDMA 传输时间安排。

证明：为了证明 DRAND 创建的 TDMA 传输时间安排是正确的，只要证明相互相距两跳以内的任意两个节点不会选择同一个时隙。这从以下几点很容易明白：

- 每个节点必须接收到其全部一跳相邻节点发送的 grant 消息；
- 相互间相距两跳的任何节点至少共享一个公共相邻节点；
- 每个节点在每轮中最多发送一条 grant 消息。

DRAND 的这个属性确保一个节点在确定其时隙时总是选择还未被其两跳相邻节点所占用的时隙号最小的那个时隙，因此两跳相邻区域内没有其他节点能够同时选择这个时隙。

3.2.4 DRAND 复杂性分析

网络中每个节点以时间周期 T_i 循环执行 DRAND。将 T_i 设为 $3d_i$ ， d_i 是节点 i 对其消息单向传输时延的估计。不要求节点同步到每轮的边界上。

为了使节点 i 完成时隙选择，所设置的 T_i 应该足够大，满足：①发送一条请求消息；②接收所有一跳相邻节点发送的同意消息；③选择可用的时隙号最小的时隙；④发送一条释放消息。开始时，每个节点 i 从 T_i 的默认值开始工作。

定义消息在网络中的单向传输时延最大值为 d_{\max} ，最小值为 d_{\min} 。令 $\Delta = d_{\max}/d_{\min}$ 、 $T_{\max} = 3d_{\max}$ 。忽略抽彩内部操作时间和时隙号选择时间，每个节点在 T_{\max} 内必须至少抽彩一次，最多抽彩 Δ 次。定义时间周期 T_{\max} 为一个超循环 (s-round)。

定理 3.2 假定最大消息时延和最小消息时延受到某些限制，那么一个节点获取一个时隙的 s-round 期望值小于 $s(\delta+1)e^{0.54}$ ，且该节点经过的循环次数 s-round 大于 s-round 期望值的 $c>1$ 倍 (c 为常数) 的概率不大于 $1/e^c$ 。

证明：令节点 j 的各个竞争节点 $C(j, k)$ 是与 j 发生冲突，且到第 k 个 s-round 循环时还没有获得自己时隙的节点的集合 (在 j 的两跳范围内)。在第 k 个 s-round 循环期间，竞争节点 $i \in C(j, k)$ 可能至少已抽彩一次。令 $L(j, k)$ 表示节点 j 在第 k 个 s-round 循环中彩。

假如节点 j 在第 k 个 s-round 循环中彩，那么节点 j 获得一个时隙，且没有其他竞争节点在本循环中彩。因为竞争节点 $i \in C(j, k)$ 在一个 s-round 循环中最多可能抽彩 Δ 次，所以节点 j 在第 k 个 s-round 循环获得一个时隙的概率 $P_r(j_{\text{leave}}, t)$ 为

$$P_r(j_{\text{leave}}, t) \geq P_r(j_{\text{leave}}, 1) \quad (3-3)$$

$$\geq P_r[L(j, 1)] \prod_{i \in C(j)} \prod_{k=1}^{\Delta} \{1 - P_r[L(i, k)]\} \quad (3-4)$$

$$\geq P_r[L(j, 1)] \prod_{i \in C(j)} \{1 - P_r[L(i, \Delta)]\}^{\Delta} \quad (3-5)$$

$$\geq \frac{1}{2(\delta+1)} \prod_{i \in C(j)} \left\{ 1 - \frac{1}{2|C(j)|+1} \right\}^{\Delta} \quad (3-6)$$

$$\geq \frac{1}{2(\delta+1)} \left\{ 1 - \frac{1}{2|C(j)|+1} \right\}^{|C(j)|\Delta} \quad (3-7)$$

$$\geq \frac{1}{2(\delta+1)} \left(\frac{1}{\sqrt{e}} \right)^{\Delta} \quad (3-8)$$

式 (3-5) 成立的原因是 $P_r[L(i, \Delta)] \geq P_r[L(i, k)]$ ，对于 $1 \leq k \leq \Delta$ 。式 (3-6) 成立的原因：

① $P_r[L(i, \Delta)] \leq 1/2[|C(j)|+1]$, 因为竞争节点 i 在设置 p_i 下使用其竞争节点集 (包含节点 j) 的最大相邻节点数的倒数 (所以最小值为 $|C(j)|+1$), 且当节点 j 选择该时隙时以及抽彩前投硬币 $1/2$ 的概率而 $|C(j)|$ 不变; ② $P_r[L(i, 1)] \geq 1/2(\delta+1)$, 因为 δ 是网络中任意节点的竞争集大小的最大值。式 (3-8) 成立的原因是 $[1-1/2(|C(j)|+1)] > 1/\sqrt{e}$ 。

由于上述结果与 j 和 k 无关, 所以给出了一个节点在任意 s -round 循环获得一个时隙的概率的下限值。令 M 表示一个节点在下限概率下, 在获得一个时隙之前所经过的 s -round 循环次数。 M 是一个随机数, 显然遵循几何分布:

$$P_r(M = k) = P_{\text{low}}(1 - P_{\text{low}})^{k-1} \quad (3-9)$$

式中, $P_{\text{low}} = 1/(2(1+\delta) \times e^{0.5\Delta})$ 。

从上面分析可以得到一个节点获取一个时隙所经历的 s -round 循环次数的期望值的上限值为

$$E[M] = 1/P_{\text{low}} = 2(1+\delta) \times e^{0.5\Delta} \quad (3-10)$$

最后, 计算节点经过 c 个 s -round 循环 (c 大于 s -round 循环次数的期望值) 仍然没有获得时隙的概率为

$$P_r[M > c \cdot E(M)] = \sum_{k=c \cdot E[M]+1}^{\infty} P_{\text{low}}(1 - P_{\text{low}})^{k-1} = (1 - P_{\text{low}})^{c \cdot E(M)} = [1 - 1/E(M)]^{c \cdot E(M)} \leq 1/e^c \quad (3-11)$$

上述分析假定消息传输时延受到某个常数的限制。实际上, 依据用来实现 DRAND 的 MAC 协议, 由于相邻节点竞争公共信道的访问, 所以节点的消息传输时延可能是其相邻区域大小的函数。在 CSMA 中可能这样, 在 CDMA 中尤其如此。但是, 对消息传输时延没有明确的限制, 这是因为消息传输时延与所使用的 MAC 协议密切相关, 对给定 MAC 协议的消息传输时延未做渐进分析。对于消息传输时延未受某个常数限制的网络, 前面的分析结果不能直接适用。下面根据实验来估计 DRAND 在这种网络中的性能。正如在 3.2.5 节看到的那样, 即使在这种网络中, 平均循环次数仍然接近分析结果。

定理 3.3 假定消息传输时延受到某个常数的限制, 那么 DRAND 的消息复杂性期望值为 $O(\delta)$ 。

证明: 在一个循环中, 一个竞争节点可以抽彩 Δ 次。在每次抽彩时, 竞争节点可以中彩和被拒绝, 因此发送 $O(1)$ 条消息。显然, 在一个循环中, 一个竞争节点可以发送 $O(\Delta)$ 条消息。由于平均需要 $O(\delta)$ 个循环, 所以每个循环过程平均可以发送 $O(\Delta \cdot \delta)$ 条消息。

3.2.5 DRAND的性能

测试床实验和 ns-2 仿真实验的目的如下: ①验证前面的分析结果, 特别是验证 DRAND 的性能随着一跳和两跳相邻节点数量的线性正比关系; ②在中等规模 Mica2 传感器测试床 (由 42 个 Mica2 节点组成) 上运行 DRAND, 评估 DRAND 的开销; ③与其他 TDMA 协议 (比如 FFRP、SEEDDEX) 对比实验, 评估 DRAND 建立的 TDMA 传输时间安排的效能。

使用 Mica2 作为实验平台, 便于软件修改 MAC 功能。对于其他电台 (比如 IEEE 802.11), 采用 ns-2 进行仿真。在 TinyOS 实现时, 使用 B-MAC 协议的默认设置 (打开 CCA、关闭 LPL 和应答), 不事先进行时钟同步。采用如下测试方案进行实验。

① 一跳 Mica2 实验: 一跳相邻区域内布置的 Mica2 无线传感器数量可变。网络中节点数可变, 从 1 个节点变化到 20 个节点, 所有节点离地面至少 2 英尺 (1 英尺=0.3048 m)。

② 两跳和多跳 Mica2 实验：图 3-7（a）表示由 42 个布置在北卡罗来纳州州立大学计算机科学大楼办公室和教室里的 Mica2 传感器节点组成的多跳无线测试床。两个节点之间的无线连接[图 3-7（a）中两个节点之间的连接线]质量变化，有些链的丢失率高达 30%~40%。两跳拓扑由两个分群和一个中心节点（节点 36）组成，每个分群 10 个节点。两个分群只能直接与中心节点通信，互为隐含终端。

③ 多跳 ns-2 仿真实验：运用 ns-2 仿真器研究 DRAND 的大规模 Ad Hoc 无线网络的可扩展性。网络拓扑由随机布置在 300 m×300 m 地面区域中的节点组成。节点传输距离 40 m，传输速率 2 Mb/s。网络相邻区域大小可变，网络节点从 50 个变化到 250 个。因此，两跳相邻区域内平均节点数在 5~60 个之间。

1. 分析结果的实验验证

（1）时间复杂性与循环次数

图 3-7（b）表示一个节点已经决定其时隙时经历的运行时间和平均最大循环次数以及该节点一个一跳相邻节点所经历的平均最大循环次数，实验结果是 10 次实验的平均结果。注意：在一跳情形下，所有节点相互处在对方的传输覆盖范围内，因此 DRAND 给每个节点分配一个唯一的时隙，这是最佳分配。

从图 3-7（b）中可以看到实验结果和前面的分析结果相吻合：所经历的循环次数随着相邻区域大小的增大而线性增大。但是，运行时间呈二次方递增。这是因为消息传输时延未受某个常数的限制，并且与网络大小有关。可见，当循环次数线性增大时，每次循环的持续时间不是恒定不变的。

（2）消息复杂性

图 3-7（c）表示在 Mica2 测试床一跳拓扑上实验得到的每个节点的消息发送量。

（3）最大时隙号

在 DRAND 中，最大时隙号受到 $\delta+1$ 的限制。但是在实际中，DRAND 分配的时隙数量可能远小于 $\delta+1$ 。图 3-7（d）表示网络密度变化条件下输入图使用的时隙数量。虚线表示 δ 。每个数据点代表 DRAND 为不同网络分配的最大时隙数量。在所有实验中，DRAND 使用的最大时隙号远小于 $\delta+1$ 。这个结果与其他时隙分配算法（如 FPRP）的性能相反，这些算法在最差情形下的时隙数量总是大于或者等于 $\delta+1$ 。

2. DRAND 开销的实验结果

（1）时间与能量的开销

测试在 Mica2 多跳网络测试床上运行 DRAND 的能耗。每个节点记录电台原始操作（接收一个字节、发送一个字节、空闲侦听）总量。根据 B-MAC 协议中列出的每种操作的能耗，对每种操作的总能耗进行加权求和即得到总能耗。

对于 Mica2 多跳网络拓扑，每个节点首先运行相邻节点寻找协议，获取其相邻区域信息；然后运行 DRAND 算法，给每个节点分配一个 TDMA 时隙；其次各个节点将其时隙信息广播给自己的两跳相邻区域；最后就可以开始使用这些时隙信息发送数据。

在测试床上对三个阶段操作（相邻节点寻找、运行 DRAND 算法、时隙分发）进行 30 次实验，其平均开销如表 3-2 所示。表 3-2 中的时间开销和能量开销是每次实验中测试床中

所有节点的平均最大值。从表 3-2 中可知，三个阶段操作的总能耗为 6.942 J，约为一个 2 500 mAh、3 V 电池（与 B-MAC 协议中使用的电池相同）的节点的总能量的 0.02%。

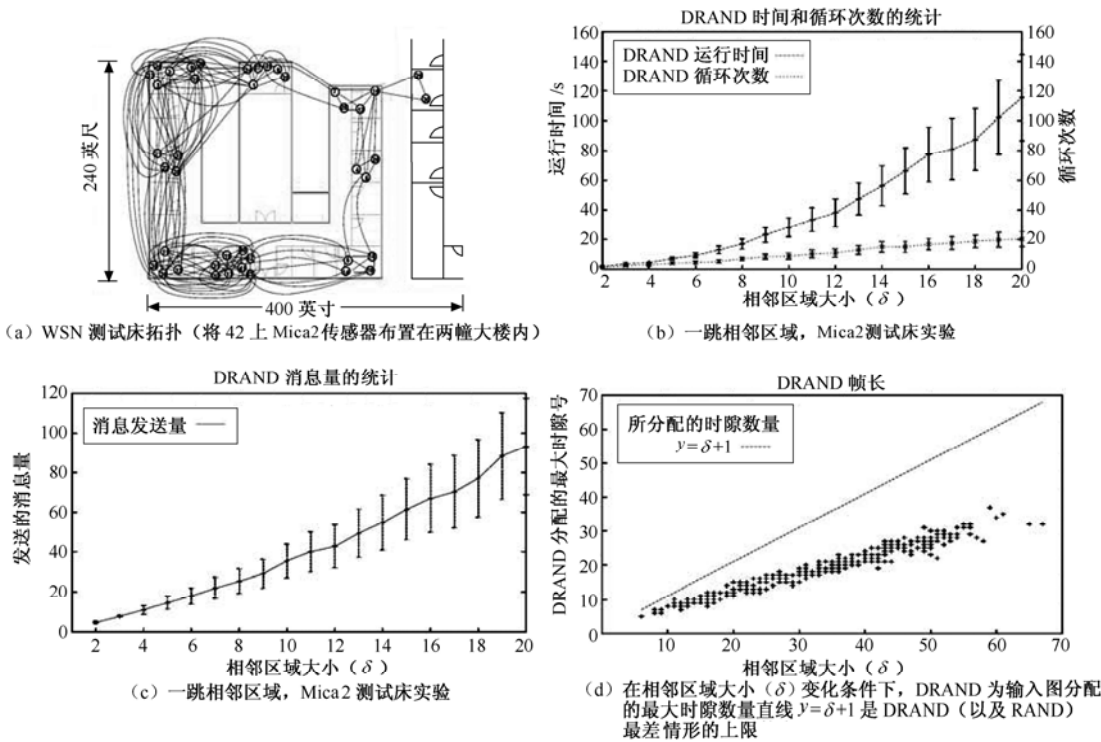


图 3-7 DRAND 分析结果的实验验证

在一次特定实验中分配给每个节点的时隙如图 3-8 (a) 所示。在图 3-8 (a) 中，每个节点上标出的两个数字分别表示该节点的 ID 以及所分得的 TDMA 时隙号。在测试床上分给节点的最大时隙号是 15。比较图 3-7 (a) 所示的网络连通性，可以验证：除少数几个节点（比如 7、12、30）以外，两跳范围内的大多数节点相互分得不同的时隙。例外的原因是存在非对称链。

尽管两跳范围内大多数节点相互分得唯一的时隙，但是也存在少数几个节点例外，如节点 7、12、30 均在同一个两跳相邻区域内，但是却分得同一个时隙 0。这是因为这些节点间的通信链为非对称链。存在非对称链时，DRAND 算法丢掉没有响应的相邻节点而继续进行。

表 3-2 DRAND 开销

操 作	平均时间/s	平均能耗/J
相邻节点寻找	30	0.732
DRAND 时隙分配	194.38	4.88
时隙分发	60	1.33
总计	284.38	6.942

(2) DRAND 恢复开销

在 DRAND 中，当一个新节点 A 加入网络（或者一个节点已分得时隙，但是由于某种原因而失效，因此重新启动）时，该节点可以通过在其一跳相邻区域内运行（或者重新运行）

DRAND 来保护一个时隙，而与 A 相距大于一跳的那些节点没有必要重新运行 DRAND。通过在测试床上重新启动一些特定节点来仿真节点入网，允许重新启动节点的一跳相邻区内的全部节点运行 DRAND，然后测试所有这些节点保护新的无冲突时隙的平均耗时和平均耗能。选择 5 个这样的节点：节点 12、19、15、2、3，这 5 个节点位于测试床上的不同区域。图 3-8 (b) 表示同时重新启动节点集 (12)、(12、19)、(12、19、15)、(12、19、15、22)、(12、19、15、22、3) 的平均耗时和平均耗能。当节点 12、19、15、2、3 全部失效时，其一跳相邻区域总共包括测试床中的全部节点，因此开销大约接近所有节点重新运行 DRAND 的开销。

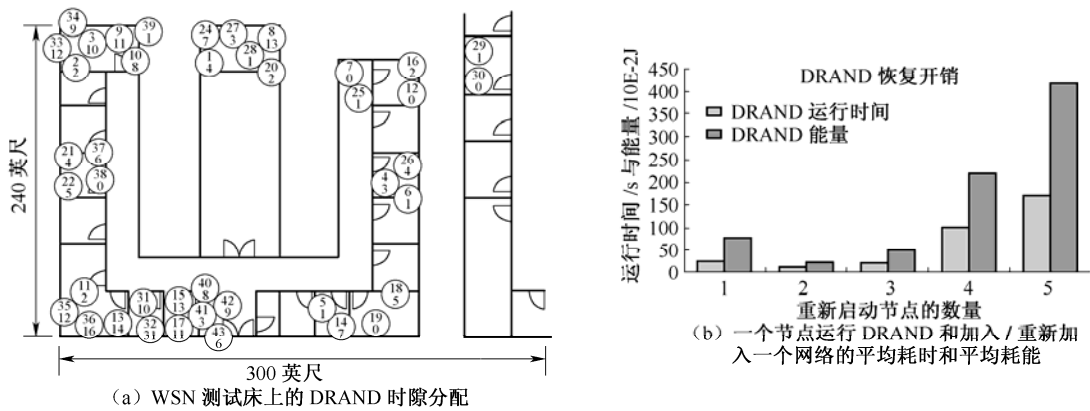


图 3-8 DRAND 的能耗

3. DRAND与现有算法的对比

(1) 比较 DRAND 与下列 MAC 协议的算法复杂性和传输效率

① FPRP：五步预留协议（Five Phase Reservation Protocol, FPRP）是一个分布式的、试探性的 TDMA 时隙分配算法。FPRP 专门设计进行动态时隙分配，将实际时间分成两个时段对的序列：预留时段和数据发送时段。在预留时段，FPRP 将下一个数据发送时段的时隙分配给有数据需要发送的节点。显然，实际上是使用前一个预留时段分配的时隙发送数据。对于数据发送时段的每个时隙，FPRP 运行五步协议，确定选择每个时隙赢取节点的循环次数。运行次数越多，FPRP 的时隙分配越好。在分布式设置下，节点不知道需要多少个循环来完成时隙分配。因此，为了与 DRAND 对比，测试在固定循环次数下 FPRP 分配的时隙数量。

② 随机 TDMA (R-TDMA)：在 R-TDMA 中，网络维护一个固定帧长 F ，设 $F=\delta$ ， δ 表示网络中两跳相邻区域的最大值。每个有分组需要发送的节点在每帧的开始从 $[0, F-1]$ 中选择一个时隙 i ， i 在 $[0, F-1]$ 上服从均匀分布，然后在该时隙发送分组。

③ SEEDDEX：在 SEEDDEX 中，在每个时隙的开头，假如一个节点有分组需要发送，那么该节点以概率 p 抽彩，假如中彩，则符合发送条件。一个节点知道其两跳相邻节点的随机数发生器的种子，因此知道其两跳相邻区域内满足发送条件的节点数量 C （包括本身），然后以概率 $1/C$ 发送。这种使用相邻节点的随机种子来确定时隙的技术叫做拓扑独立传输时间安排技术。

(2) 算法复杂性对比

图 3-9 (a) 至图 3-9 (c) 表示 DRAND、FPRP 的实验结果，考虑的对比性能是每种算法

的最大时隙号、每个节点的消息发送量、总的运行时间。在多跳网络拓扑以及 ns-2 仿真环境下进行实验，平均两跳相邻节点从 8 个变化到 52 个。SEEDEx 和 R-TDMA 没有最大时隙号概念，两者连续运行。因此没有针对 SEEDEx 和 R-TDMA 作该性能对比实验，而是比较其传输效率。

图 3-9 (a) 至图 3-9 (c) 中的 FPRP- x 表示 FPRP 运行 x 个循环才找到获得每个时隙的赢取节点。随着相邻区域的增大，FPRP 的最大时隙号、每个节点的消息发送量、总的运行时间也随着增大。DRAND 的时隙数量、消息复杂性总是优于 FPRP。DRAND 要求的时隙总是小于 FPRP (少 34%)。因为时隙数量代表 TDMA 帧的长度，所以时隙数的减少量转化为同等量的总信道利用率的提高。DRAND 的运行时间与 FPRP-30 相当，但是小于 FPRP-50。考虑到 DRAND 的时隙分配效率比 FPRP-50 高许多，所以这个结果令人鼓舞。DRAND 的消息发送量远低于 FPRP，这就意味着 DRAND 的能耗比 FPRP 低得多。

(3) 传输效率对比

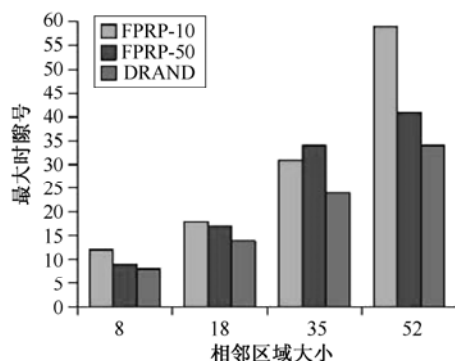
现在比较 DRAND、SEEDEx、R-TDMA 建立的时隙分配对吞吐量的影响。前面已经通过实验说明 FPRP 产生的时隙多于 DRAND，因此这里就不纳入 FPRP 来进行吞吐量比较。在 TinyOS 中实现 TDMA 协议，以便在 Mica2 传感器节点上进行吞吐量比较实验。每个节点只是在其自己分得的时隙上发送分组。比较 DRAND、SEEDEx、R-TDMA 分别选择的时隙下的吞吐量，在如图 3-7 (a) 所示的测试床上进行实验和测试。

在开始发送数据前，采用 Z-MAC 协议的本地时间同步法确保所有节点时隙同步。尽管时间同步，但是由于同步消息丢失以及同步间的时钟漂移，节点可能失步。因此，确保节点在每个时隙开始前的 5 ms 内不发送。5 ms 作为同步错误的缓冲时间，有助于防止由于同步错误而导致在时隙边界发生碰撞。Mica2 电台 (CC1000) 以 19 200b/s 的无线数据传输速率发送一个 46 B 分组 (10 B 的 MAC 前导和同步，36 B 的有效载荷) 约需要 19.2 ms。因此，将一个 TDMA 时隙长度设为 25 ms，足够满足发送一个分组和缓冲时间。

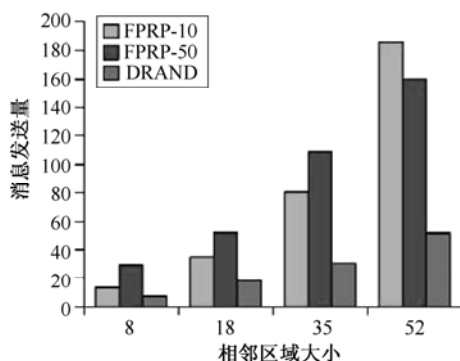
在测试床上对一跳和两跳的 Mica2 网络拓扑进行实验。对于 DRAND、SEEDEx、R-TDMA，一跳结果只是稍好于两跳结果 (由于不存在隐含终端问题)，因此这里未做报告。对于 SEEDEx，根据文献研究分析，以相邻区域大小分别为 6、12、21 个节点的最佳概率 $p=0.246$ 、0.117、0.074 进行实验。图 3-9 (d) 表示中心节点吞吐量随着发送节点的增多的变化情况。 $p=0.246$ 、0.117 的性能结果差于 $p=0.074$ ，所以没有给出 $p=0.246$ 、0.117 的实验结果。从实验中发现：确定性 TDMA (比如 DRAND) 的吞吐量总是优于随机性 TDMA (比如 SEEDEx、R-TDMA)，其原因有两点：①随机性 TDMA 导致碰撞概率较高 (高达 20% 以上)；②随机性 TDMA 趋向于平均发送较少的分组。

通过改变随机性 TDMA 的特定参数，有可能提高其消息发送量。例如，调整 p 小于 0.074，随机性 TDMA 的消息发送量有可能高于 DRAND，但同时又提高了碰撞机会，因此导致吞吐量下降。 p 取最佳值，确保在特定发送速率下 SEEDEx 的吞吐量最高。

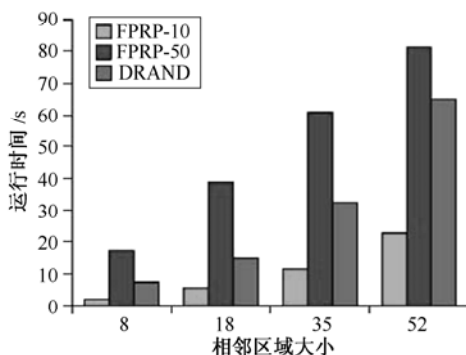
在图 3-9 (d) 中，采用如前所述的 WSN 测试床，两个分群，每个分群 10 个节点。每次运行时分别从每个分群中随机选择一个发送节点，发送节点从 1 个变化到 20 个。所有节点总是准备有分组需要发送。



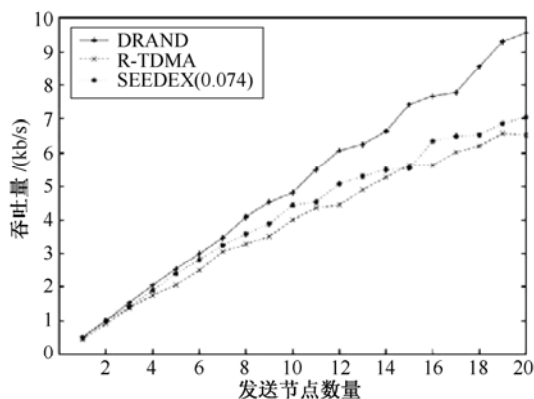
(a) DRAND、FPRP 的最大时隙号对比



(b) DRAND、FPRP 的消息发送量对比



(c) DRAND、FPRP 的平均耗时对比



(d) DRAND、R-TDMA、SEEDEX 的两跳吞吐量对比

图 3-9 DRAND 与其他 MAC 协议的性能对比

3.3 功率高效与时延意识媒介访问协议 (PEDAMACS)

美国加州大学伯克利分校研发的 WSN 功率高效与时延意识媒介访问协议 (Power Efficient and Delay Aware Medium Access Protocol, PEDAMACS) 是能量效率极高的 WSNMAC 协议。PEDAMACS 寻找网络拓扑, 维持节点同步, 正确执行 TDMA 传输时间安排; PEDAMACS 适用于具有以下一个特点的多跳无线网络: 所有数据分组传输给同一个节点 (将其称为访问点, 即中心节点), 访问点 (Access Point, AP) 具有足够功率对其所有一跳相邻节点进行发送。对于这种网络, 就网络寿命和有保证时延而论, PEDAMACS 的表现优于现有随机访问协议。

3.3.1 PEDAMACS 协议概述

(1) PEDAMACS 协议的假设条件

PEDAMACS 的正确执行需要发射功率、数据流量、链路条件、路由满足以下条件:

① 无线网络由一个 AP 和若干个传感器节点组成。根据需要, AP 能够访问其所有一跳相邻节点。每个传感器节点能够调整其发射功率。一个传感器节点需要经过多跳路径到达 AP。

② 传感器节点周期性产生数据，每个传感器节点的数据产生速率可能不相同，产生的数据均要传递给 AP。由此很容易延伸到 WSN 中只有一些传感器节点给 AP 转发数据。

③ 双向链路。正确完成网络功能（比如分布式 Bellam-Ford 路由算法）需要双向链路。假如所有传感器节点以相同功率进行发送，则可实现链路的双向性。根据接收信号强度设置链路，可以补偿硬件差异引起的实际发射功率差异。

④ 传感器节点低速移动。对于高速移动，PEDAMACS 的时延表现较差。

⑤ 任何基于其他最低链路开销参数的路由协议（比如最大寿命路由协议）可以与 PEDAMACS 协议一起使用。

(2) PEDAMACS 协议的操作

PEDAMACS 协议操作分成 4 个阶段：传感器节点本地拓扑建立阶段、AP 拓扑信息收集阶段、传输时间安排阶段和拓扑调整阶段。在本地拓扑建立阶段，每个传感器节点识别其本地拓扑，即其相邻节点、干扰节点以及其路由树上的父节点（根据某些路由参数建立的以 AP 为树根的路由树上）等。在拓扑信息收集阶段，每个传感器节点将其本地拓扑信息发送给 AP，本阶段结束时，AP 获得整个网络的拓扑。在传输时间安排阶段开始的时候，AP 广播一个传输时间安排。然后每个传感器节点遵循这个传输时间安排，在其他节点发送间隙时进行休眠或者侦听信道。拓扑调整阶段在必要之时才被触发和启动，用于寻找网络中的变化，以及用于获取在拓扑建立阶段没有获取的拓扑信息。

(3) 发射功率等级与传输距离

PEDAMACS 节点采用三个发射功率等级 $P_l > P_m > P_s$ ，对应的传输距离分别为 $r_l > r_m > r_s$ 。只有 AP 采用最大发射功率等级 P_l 给其所有一跳相邻传感器节点广播协调分组。所有传感器节点采用最低发射功率 P_s 将其数据分组沿着多跳路径转发给 AP。尽管 P_s 小而可以降低功耗，但是 P_s 必须足够大，以便维护网络的连通性。传感器节点采用中级发射功率 P_m 进行发送，以便寻找其本地拓扑。

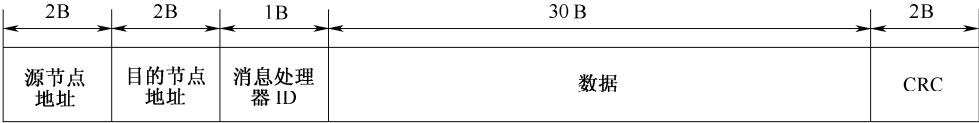
当一个传感器节点以最低发射功率 P_s 发送其数据分组时，有些传感器节点能够以可承受的比特误码率（BER）接收和解析该数据分组，将这些传感器节点称为该节点的相邻节点，而另外一些传感器节点接收到该数据分组时信号太弱而不能解析该数据分组，但是其接收信号却强到能够干扰其他信号，将这些传感器节点称为该节点的干扰节点。一个传感器节点的本地拓扑由其相邻节点和干扰节点组成。

3.3.2 PEDAMACS 分组格式

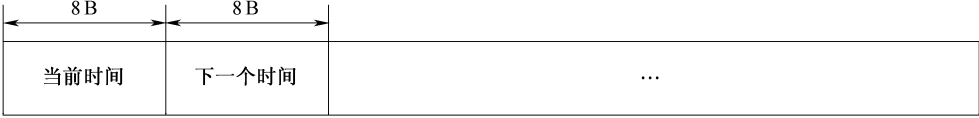
图 3-10 给出了 PEDAMACS 分组结构。TinyOS 的基本分组结构：分组头为 5 B，数据载荷为 30 B，CRC 为 2 B。

3.3.3 本地拓扑建立阶段

当 AP 以最大发射功率等级 P_l 给所有传感器节点广播一个拓扑建立（Topology Learning）协调分组的时候，就开始进入各个传感器节点本地拓扑建立阶段。这个协调分组包含当前时间（Current Time）以及 AP 下一个协调分组发送时间（Next Time）。所有传感器节点同步到该分组携带的当前时间上，停止发送，在 AP 下一个协调分组发送时刻侦听 AP 发送的协调分组。



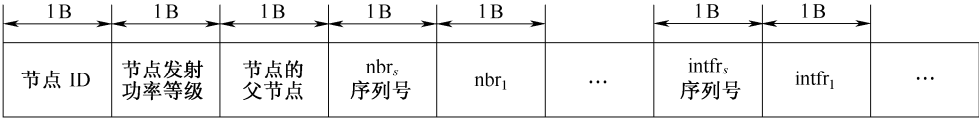
(a) TinyOS 分组结构



(b) 协调分组



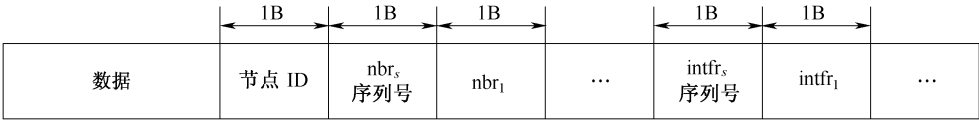
(c) 路由树结构分组



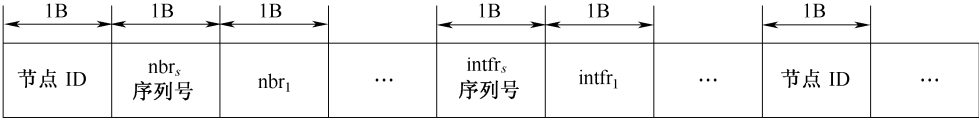
(d) 本地拓扑分组



(e) 传输时间安排分组



(f) 数据分组



(g) 拓扑调整分组

图 3-10 PEDAMACS 分组结构

AP 广播完拓扑建立协调分组后，采用中级发射功率 P_m 在网络中泛洪一个树结构（Tree Construction）分组。树结构分组包含路由树上发送节点的开销（比如到达 AP 的最小转发跳数）。传感器节点接收到树结构分组后，首先按照其干扰模型，根据接收信号强度确定树结构分组是来自自己的某个相邻节点还是来自某个干扰节点。假如发送节点是接收节点的一个相邻节点，并且是路径上的下一个转发跳（该路径的开销低于先前所建立的路径），那么接收节点更新这个开销——填入自己的开销，然后重播这个树结构分组；同时接收节点按照数组方式保存这个相邻节点或者干扰节点与其开销和接收信号强度相关的信息。在泛洪结束的时候，接收节点选择其父节点作为到达 AP 最低开销路径上的下一个转发跳。

PEDAMACS 可以采用任何干扰模型。这里举例说明一个干扰模型。分组成功接收的条件：信号与干扰、噪声之和的比率（Signal-to-Interference-plus-Noise Rate, SINR）大于某个

门限值 β , β 跟可承受 BER、检波器结构、调制/解调方案、信道编码/解码算法等有关。另一方面, SINR 依赖信道、干扰、天线增益、发射功率。从节点 i 到节点 j 的最短传输距离的 SINR 为

$$\text{SINR}_{ij} = P_{r,s}^{ij} / (I_{m,j}^i + I_{l,j}^i + \sigma^2) \quad (3-12)$$

式中, $P_{r,s}^{ij}$ 表示节点 j 接收节点 i 以最低功率 P_s 发送的接收功率, σ^2 表示接收机热噪声功率, $I_{m,j}^i = \sum_{k \neq i, j, |d(k,j)|, r_m} P_{r,s}^{kj}$ 表示除节点 i 以外的其他发送节点 (处在节点 i 的中级发射功率传输覆盖范围内) 在节点 j 的干扰功率, $I_{l,j}^i = \sum_{k \neq i, j, |d(k,j)| > r_m} P_{r,s}^{kj}$ 表示除节点 i 以外的其他发送节点 (处在节点 i 的中级发射功率传输覆盖范围外) 在节点 j 的干扰功率, $|d(k,j)|$ 表示节点 k 与节点 j 之间的距离。

采用中级发射功率 P_m 允许传感器节点确定传输覆盖范围内的干扰节点, 删除 $I_{m,j}^i$ 项。总干扰 $I_{\text{total},j}^i = I_{m,j}^i + I_{l,j}^i$ 对于相同的节点配置是恒定的。因此, 系统使用的发射功率比率 P_m/P_s 越大, $I_{m,j}^i/I_{l,j}^i$ 越大, 因而分组成功接收的概率越高, 但是由于系统中干扰节点随着增多, 因而系统经历的时延随着增大。

由于中级发射功率下的接收功率 $P_{r,m}^{ij}$ 跟最低发射功率下的接收功率 $P_{r,s}^{ij}$ 有关, 即 $P_{r,s}^{ij} = P_{r,m}^{ij} \times (P_s/P_m)$, 所以从节点 i 到节点 j 的最短传输距离的 SINR 为

$$\text{SINR}_{ij} = (P_{r,m}^{ij} \times (P_s/P_m)) / (I_{l,j}^i + \sigma^2) \quad (3-13)$$

假如中等传输距离大到足够覆盖几乎所有干扰节点, 那么可以忽略 $I_{l,j}^i$ 项, 根据树结构分组的接收信号强度 $P_{r,m}^{ij}$ 计算 SINR_{ij} 。若 $\text{SINR}_{ij} > \beta$, 则节点 i 和节点 j 在最短传输距离下是相邻节点; 否则, 节点 i 和节点 j 是干扰节点。

选择足够大的 β 对于提供可靠路由非常重要, 这是因为在传输时间安排阶段安排各个传感器节点的发送时间, 没有重传。为了得到 β 估计值, 将伯克利 Mica2dot 传感器间的接收成功率与链路非对称性作为接收信号强度的函数来测量。图 3-11 (a) 表示接收信号强度位于某个门限值 (在图 3-11 (a) 中为 -85 dBm) 之上时成功接收概率达 95% 以上。图 3-11 (b) 表示不同接收信号强度下的链路非对称性。按照节点 i 和节点 j 的接收成功概率之绝对差测量链 (i, j) 的非对称性。从图 3-11 (b) 中观察到: 假如接收信号强度大于某个特定值 (在图 3-11 (b) 中为 -80 dBm), 那么可以认为链路是对称链。因为在本项实验中没有干扰节点, 所以比较 SINR 与 β 等效于比较中等传输距离下接收信号强度与另一个门限值

$$P_{r,m}^{ij} \geq \alpha \times (P_m/P_s) \quad (3-14)$$

式中, $\alpha = \beta \sigma^2$ 等于最短传输距离下链路质量优良的门限值 [在图 3-11 (b) 中为 10^{-8} mW]。

在拓扑建立阶段采用随机访问 MAC 协议, 因为此时传感器节点还没有传输时间安排。在完成拓扑建立阶段的时候, 几乎所有传感器节点都能够高概率地确定自己的相邻节点和干扰节点。采用类似于 IEEE 802.11 的载波侦听多址访问 (CSMA) 协议。传感器节点首先侦听信道一段随机时间, 若信道侦听为空闲才进行发送。在载波侦听前增加一段随机时延, 以便进一步减轻碰撞。

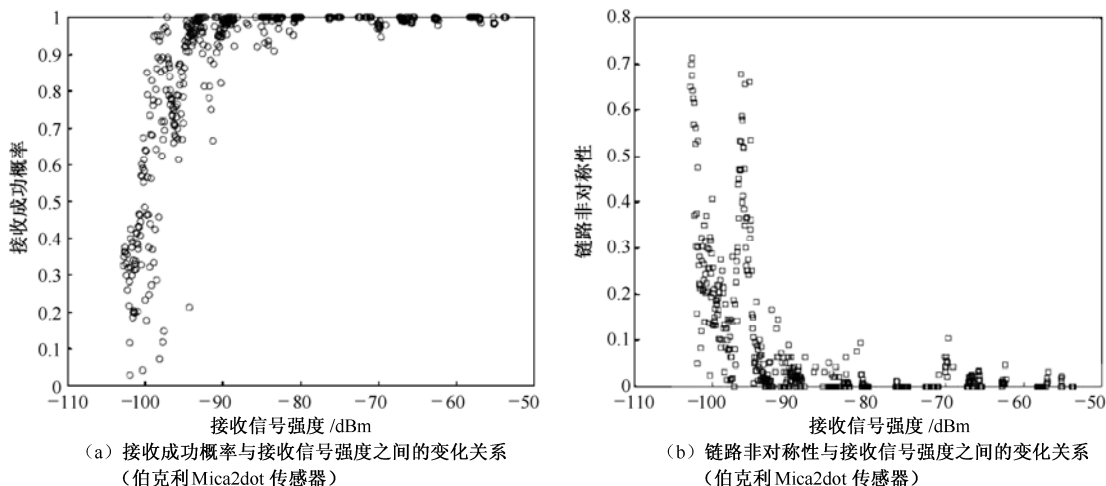


图 3-11 接收信号强度对接收成功率和链路非对称性的影响

3.3.4 AP 拓扑信息收集阶段

当拓扑信息收集阶段结束的时候，AP 具有完整的全网拓扑信息。AP 在其广播的拓扑建立协调分组中指定的下一个时间广播一个拓扑收集分组，开始拓扑信息收集阶段。拓扑收集分组也包括当前时间（用于传感器节点时间同步）以及 AP 下一个协调分组发送时间。

每个传感器节点接收到 AP 发送的拓扑收集分组后，采用最低发射功率 P_s 将其本地拓扑 (Local Topology) 分组发送给自己的父节点，本地拓扑分组包含该传感器节点的父节点、相邻节点、干扰节点。

拓扑收集阶段也采用 CSMA。但是，采用 CSMA 引起的碰撞至少在两个传感器节点上导致本地拓扑信息受损，CSMA 本身不能保证 AP 接收到完整的全网拓扑信息。因此，对 CSMA 进行改进，使其包含隐含确认：一个传感器节点接收到其父节点对其父节点的发送时就包含确认之意，第一级传感器节点发送的分组由 AP 直接确认——AP 重传所接收到的所有分组或者像 IEEE 802.11 那样直接发送一个确认。

3.3.5 传输时间安排阶段

AP 根据自己收集到的全网拓扑信息直接安排所有传感器节点的发送时间。将传输时间安排帧划分成时隙。一个时隙的长度等于一个分组传输时间加上一个保护间隔，保护间隔用于补偿时间同步误差。

在传输时间安排阶段开始的时候，AP 广播一个传输时间安排 (Scheduling) 分组。与其他协调分组一样，传输时间安排分组也包括当前时间（用于传感器节点时间同步）以及 AP 下一个协调分组发送时间，另外还包括 AP 做出的各个传感器节点的传输时间安排。

每个传感器节点在传输时间安排帧开始的时候产生数据分组，采用最低发射功率 P_s 将数据分组发送给 AP。数据分组包含需要发送给 AP 的数据以及新的拓扑信息，新拓扑信息包括自最近一次拓扑建立阶段和拓扑信息收集阶段以来在拓扑调整阶段发现的该节点的相邻节点和干扰节点。假如一个数据分组不能承载全部新拓扑信息，则按照循环方式由每个数据分组

共同承载新拓扑信息。数据分组数据域的长度依赖 WSN 的具体应用。

传输时间安排确定传感器节点的发送时隙。一个传感器节点接收到一个分组后，不是立即转发该分组，而是将分组送入队列排队，然后休眠，直到到达自己发送时隙的时候才开机发送。这个传输时间安排算法确保所有分组在本阶段结束的时候全部传递到达 AP。稍后将详细描述 PEDAMACS 的传输时间安排算法。

保护间隔只占时隙长度的一小部分。因为 AP 给所有传感器节点发送相同分组，所以排除了时戳和发送节点分组发送时刻之时延引起的时间同步误差。因为 AP 的传输距离为几百米，所以也可以忽略传播时延（几个微秒）。假定所有传感器节点运行相同软件，那么所有传感器节点在相同时刻给分组加时戳。因此，应用中的唯一时间同步误差来源于时钟脉冲相位差，即各个传感器节点的时钟嘀嗒速率之差。传感器节点的典型时钟漂移是 1 s 内 30~50 μ s。假定每个传感器节点的分组产生周期是 30 s，那么最大时钟漂移是 0.9~1.5 ms，而以 50 kb/s 传输速率发送一个 37 B 的 TinyOS 分组的时间为 14 ms。AP 还可以在本阶段开始与结束之间发送其他协调分组，以便进一步缩短保护间隔。

3.3.6 拓扑调整阶段

在完成传输时间安排阶段后进入拓扑调整阶段，以便识别完整的网络拓扑、检测传感器节点移动和根据应用需要添加的新传感器节点。假如没有成功安排所有节点的传输时间，则意味着 AP 由于拓扑信息错误而在传输时间安排阶段给几个相互冲突的传感器节点安排了一个时隙。但是，假如相互冲突传感器节点很少，那么重新启动拓扑建立阶段可能会引起已成功分得发送时隙的节点的分组时延。拓扑调整阶段有助于 PEDAMACS 不用重新启动拓扑建立阶段，因拓扑发生微小变化而更新传输时间安排。成功分得发送时隙的传感器节点所占比例下降的另一个原因可能是传感器节点与其父节点之间的通信链路不可靠，其处理方法是在网络中产生冗余数据，采用多条路径发送数据。

当 AP 给所有传感器节点广播一个调整 (Adjustment) 协调分组的时候，就开始拓扑调整阶段。调整协调分组也包括当前时间（用于传感器节点时间同步）以及 AP 下一个协调分组发送时间。

将拓扑调整阶段所占时间减去分组发送时间和保护间隔时间得到退避窗口的时间长度，即退避窗口大小等于 Next Time 与 Current Time 之差。传感器节点等待一段从退避窗口大小中随机选择的时间，等待时间结束后信道若是空闲则发送其调整拓扑 (Adjustment Topology) 分组，调整拓扑分组包含新拓扑信息。其间，传感器节点可以接收其他传感器节点发送的分组。

拓扑调整阶段采用中级发射功率 P_m ，以便检测干扰节点和相邻节点。传感器节点若是在本阶段检测到新的干扰节点或者相邻节点，则将这些信息填入其在传输时间安排阶段发送的数据分组中，也可以将这些信息填入其在下一个拓扑调整阶段发送的调整拓扑分组中，以便保证信息传递到达能够在传输时间安排阶段将数据分组成功发送给 AP 的传感器节点。在必要的时候，AP 可以根据该信息更新路由和（或者）修改传输时间安排。

3.3.7 传输时间安排算法

AP 收集到全部拓扑信息后，执行传输时间安排算法。该算法给一组无冲突传感器节点

分配发送时隙，因此当传输时间安排阶段结束的时候，每个传感器节点发送的数据分组传递到达 AP。

1. 网络与传输模型

在传输时间安排阶段开始时，WSN 可以表示为一张图 $G=(V, E)$ ， V 表示传感器节点集合（包括访问点 AP），边（无方向） $E \subset V \times V$ 表示需要安排传输时间的通信链路，图构成一棵树，树根在 AP。所有流量都是传给 AP 的，所以传感器节点的每个数据分组都要转发给自己的父节点。

一个传感器节点可能干扰另一个传感器节点，所以这类传感器节点不该同时发送。假定已知干扰图 $C=(V, I)$ （在传输时间安排阶段开始时）， $I \subset V \times V$ 表示这样的边集合：假如传感器节点 u 和 v 能够相互旁听到对方或者其中一个传感器节点（ u 或者 v ）能够干扰发送给其他一个传感器节点的信号（即使这两个传感器节点不能相互旁听到对方），则边 $(u, v) \in I$ 。因此，假如 u 正在发送，那么不应该安排 v 此时接收其他传感器节点的发送。所以 I 由每个传感器节点与其相邻传感器节点和干扰节点之间的通信链路组成（不包括与其父节点之间的通信链路）。

与图 $G=(V, E)$ 和干扰图 $C=(V, I)$ 对应的冲突图是图 $G_C=(V, E_C)$ ，其中 E_C 由不该同时发送的传感器节点对之间的边组成。 E_C 包含两种类型的边：第一，若 $(i, j) \in E$ ，则边 $(i, j) \in E_C$ ，这是因为一个父节点和一个子节点（即一对父子传感器节点）不能同时发送；第二，若 $(i, j) \in I$ 或者 $(i, j) \in E$ 并且 c_j 是 G 中 j 的一个子节点，则 $(i, c_j) \in E_C$ ，这是因为 i 和 j 相互干扰，假如 i 正在发送，那么由于 j 能够旁听到 i 和 c_j 的发送，因而 j 的子节点 c_j 不能同时发送。图 3-12（a）表示图 G 、 C 、 G_C 之间的关系。

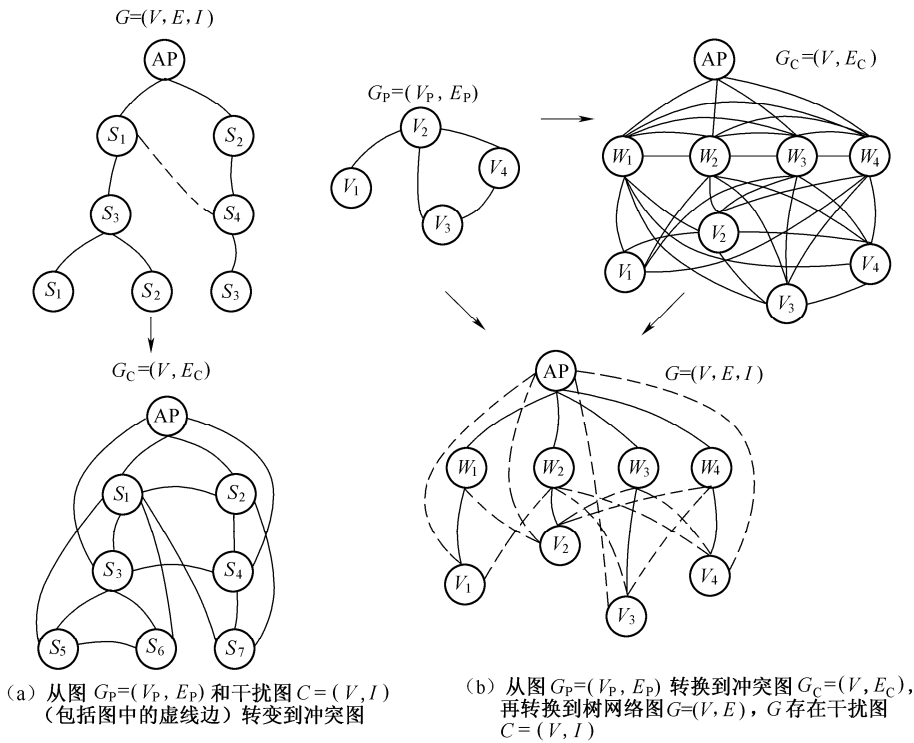


图 3-12 PEDAMACS 图转换

传输时间安排帧的长度等于从传输时间安排协调分组结束之时到传输时间安排阶段结束之时之间的时间长度。因此，当每个传感器节点严格产生了一个分组的时候，就开始传输时间安排帧；而当所有这些分组传递到达 AP 时，就结束传输时间安排帧。

将传输时间安排帧划分成时隙。一个时隙足够发送一个数据分组以及补偿时间同步误差的保护间隔。一个传输时间安排给 E 中的每条边分配一个或者多个时隙。传感器节点 u 可以在分配给边 $(u, v) \in E$ 的时隙中接收其子节点 v 发送的分组。

采用如下符号：传感器节点 u 和 v 之间的距离表示为 $d(u, v)$ ，等于 u 和 v 在图 G 之间的传输路径上的边数；假如传感器节点 u 与 AP 之间的距离为 k ，则称 u 处在 k 级。

2. 传输时间安排问题

图 G 中的每个传感器节点在传输时间安排帧开始时产生一个分组。给定干扰图 C ，传输时间安排问题就是寻找最短的帧，所有传感器节点在该帧期间能够将其分组发送给 AP。

定理 3.4 传输时间安排问题是 NP-完全问题。

证明：将求一张图颜色数的 NP-完全问题简化为传输时间安排问题。假如 G 的顶点可以添加 k 种不同的颜色，且相邻顶点颜色不相同，则称 G 是 k 可着色。图 G 的颜色数等于最小数 k ，使得 G 是 k 可着色。设 $G_P=(V_P, E_P)$ ，且 $V_P=\{v_1, v_2, \dots, v_N\}$ 是一张要求其颜色数的图例。首先构建冲突图 $G_C=(V, E_C)$ ：第一， G_C 包括 G_P 的所有传感器节点和所有边；第二，对每个传感器节点 v_i 增加另一个传感器节点 w_i ；第三，对所有 i, j ；添加边 $(w_i, w_j) \in E_C$ ；第四，对所有 i 添加一个新的 AP、边 (AP, w_i) 。结果见图 3-12 (b)。

冲突图 G_C 满足：若 w_i 是活动传感器节点，则 $V \setminus \{w_i\}$ 中没有同时活动的传感器节点；若 v_i 是活动传感器节点，则根据边 E_P ， V 中没有任意活动的 w_j 或者冲突节点。

现在构建一棵树 $G=(V, E)$ 、一张干扰图 $C=(V, I)$ ，其中冲突图 $G_C=(V, E_C)$ 。树的边 $E=\{(AP, w_i), (w_i, v_i) | 1 \leq i \leq N\}$ 。

因为 AP 是 w_i 的父节点，所以 $(w_i, AP) \in E_C$ ，对所有 i ；且由于 w_i, w_j 具有同一个父节点 AP，因而 $(w_i, w_j) \in E_C$ ，对所有 i, j ；由于 w_i 是 v_i 的父节点，所以 $(v_i, w_i) \in E_C$ 。

设 I 包含边 (v_i, AP) 、 (v_i, w_j) 、 (v_j, w_i) ，其中包括所有 i, j ，且 $(v_i, v_j) \in E_C$ 。

因为 $(v_i, AP) \in I$ 、 $(w_j, AP) \in E$ ，所以 $(v_i, w_j) \in E_C$ ，对所有 i, j 。若 $(v_i, w_j) \in I$ 、 $(v_j, w_i) \in I$ ， $i \neq j$ ，由于其中一个传感器节点的父节点受到另一个传感器节点的干扰，所以 $(v_i, v_j) \in E_C$ 。因此， G_C 确实是树图 G 和干扰图 C 的相应冲突图。

考虑 G_C 的最小传输时间安排表长度，每个传感器节点 $v_i, w_i, 1 \leq i \leq N$ ，有一个分组发送给 AP。 w_i 的分组采用路径 (w_i, AP) ， v_i 的分组采用路径 (v_i, w_i, AP) 。由于每个 w_i 与传感器节点 w_j 和所有传感器节点 v_i 冲突， $i \neq j$ ，所以采用 N 个时隙将一级传感器节点（离 AP 一跳远的传感器节点）产生的分组发送给 AP，而与网络其他部分无关。当 N 个分组从二级传递到一级的时候，则采用另外 N 个时隙将其转发给 AP。

因此，为了将所有分组发送给 AP 所需时间最少，则必须分组从二级传递到一级所需时间最少。但是二级冲突图由原图 G_P 决定，所以做出传输时间安排所需的最少时间严格等于 $2N+c$ ， c 表示原图 G_P 的颜色数。

定理 3.5 最小传输时间安排表长度至少等于 $|V|-1$ 、最大等于 $((|V|-1)|V|)/2$ ， $|V|$ 表示 V 中的传感器节点数。

证明：AP 在每个时隙中最多能够接收一个分组，所以将所有分组传递给 AP 至少需要 $|V|-1$ 个时隙，这是下限值。最差干扰图是完全的：任何一个传感器节点的发送干扰所有其他传感器节点。在这种情况下，在一个时隙中最多可以发送一个分组（在网络内）。在直线网络（最差情形）中，每个父节点最多一个子节点，所有分组必须传递经过的总转发跳数等于

$$1+2+\cdots+|V|=[(|V|-1)|V|]/2$$

3. PEDAMACS传输时间安排算法

由于许多非冲突传感器节点子集候选每个时隙，为一个发送时隙选定的非冲突传感器节点子集影响下一个时隙的发送数量，有些能够安排传输时间的传感器节点因为在前一个时隙中选定的非冲突传感器节点子集而可能不能发送分组，所以传输时间安排是个复杂问题。下面描述一个保证传输时间安排帧长度范围的多项式时间算法。

PEDAMACS 传输时间安排算法有三个组成部分：①从原始网络推导出直线网络 $G_L=(V_L,I_L)$ ，由此得到冲突图 $G_{CL}=(V_L,E_{CL})$ ；②给直线网络添加颜色，具有相同颜色的传感器节点组成一个最大独立集 G_{CL} ；③根据直线网络的添色，安排原始网络中各条通信链路的传输时间， $(u,v)\in E$ 。

4. 直线网络

假如原始树网络深度为 N ，那么直线网络 $G_L=(V_L,I_L)$ 有传感器节点 $V_L =\{v_1,v_2,\cdots,v_N\}$ 和边 $(v_i,v_{i+1})\in E_L, 1\leq i\leq N, v_l$ 对应原始网络中 l 级的所有传感器节点。若原始网络中一个 j 级节点、一个 l 级节点之间存在一条干扰边，则干扰图 $G_L=(V_L,I_L)$ 包含边 $(v_j,v_l), j, l\geq 1$ 。因此，若原始网络中一个 j 级节点与一个 l 级节点的发送发生冲突，则得到的冲突图 $G_{CL}=(V_L,E_{CL})$ 包含边 $(v_j,v_l), j, l\geq 1$ 。图 3-13（a）给出了求图 E_L, C_L, E_{CL} 的算法，算法运行时间为 $O(|V|^2)$ 。

5. 直线网络的添色

给直线网络添色，具有相同颜色的两个传感器节点不能在相同时间发送。添色算法包含两个阶段：第一阶段给每个传感器节点分配一种颜色，如图 3-13（b）所示；第二个阶段检查每种颜色的所有传感器节点是否可以分得这同一种颜色，从而保证具有相同颜色的传感器节点构成一个最大无冲突节点集，如图 3-13（c）所示。

第一阶段给传感器节点 i 分配一个时隙需要 $O(i)$ 步操作，所以颜色分配算法运行时间为 $O(|V|^2)$ 。假如使用 M 个颜色，那么第二阶段的运行时间为 $O(|V|^2M)$ 。

6. 安排原始网络的传输时间

假如直线网络中两个传感器节点 $v_i、v_j$ 分得同一种颜色，则 $v_i、v_j$ 不会相互干扰、并且能够同时发送。通过构建直线网络，原始网络中 i 级传感器节点和 j 级传感器节点之间不存在干扰，所以从 i 级任选一个传感器节点和从 j 级任选一个传感器节点可以同时发送。

一个超时隙就是连续几个时隙的组合，树上每级在超时隙开始时至少有一个分组，在超时隙期间至少将一个分组转发给下一级。由于不同级上的两个传感器节点尽管分得同一种颜色，但是可以同时发送，所以一个超时隙包含的时隙数最多等于直线网络添色使用的颜色数。

根据直线网络颜色确定了当前时隙对应的等级后，选择处于各个等级的无冲突节点集进行发送（有分组需要发送）。无冲突节点集可能正好包含一个传感器节点或者节点组，只要这些节点间的边不属于冲突图 G_C 。

原始网络传输时间安排算法如图 3-13（d）所示。假如从对应每种颜色的每个时隙的等级上选择一个传感器节点，那么运行时间为 $O(I)$ ， I 表示传输时间安排帧中的时隙数。

```
Input: (V,E,I,EC).
Output: (VL,EL,IL,ECL).
begin
    add node  $v_1$  to VL
     $I = 2$ 
    while  $I \leq levelOfTree$ 
        add node  $v_1$  to VL
        add edge  $(v_{l-1}, v_l)$  to EL
        If  $\nexists (u,v) \in I(EC)$  with  $u$  at level  $l$  and  $v$  at level  $j$  satisfying  $j < I$ 
            add edge  $(v_j, v_l)$  to IL(ECL)
         $l++$ 
    end
```

（a）与原始网络对应的直线网络的寻找算法

```
Input: VL = { $v_1, v_2, \dots, v_N$ }, graph
GL = (VL,EL) with conflict graph
GCL = (VL,ECL).
Output: One color assigned to each node
{( $v_1, c_{v1}$ ), ( $v_2, c_{v2}$ ), ..., ( $v_N, c_{vN}$ )} in which  $c_{v_i} \in \{1, 2, \dots, M\}$  and  $M$  is the number of colors.
begin
    for  $l = 1$  to  $N$ 
         $s = 1$ 
        while (there is a node conflicting with  $v_l$  with color  $s$ )
             $s = s + 1$ 
        assign color  $s$  to  $v_l$ 
    end
```

（b）给直线网络中的每个传感器节点分配一种颜色

```
Input: VL = { $v_1, v_2, \dots, v_N$ }, graph
GL = (VL,EL) with conflict graph
GCL = (VL,ECL), one color assigned to each node {( $v_1, c_{v1}$ ), ( $v_2, c_{v2}$ ), ..., ( $v_N, c_{vN}$ )} in which  $c_{v_i} \in \{1, 2, \dots, M\}$ .
Output: Color assignment to each node such that each color corresponds to a maximal nonconflicting set.
begin
    for  $s = 1$  to  $M$ 
        for  $i = 1$  to  $N$ 
            if (no node with color  $s$  conflicts with  $v_i$ )
                add color  $s$  to the color set of  $v_i$ 
        end
    end
```

（c）给直线网络中的每个传感器节点分配多种颜色

图 3-13 PEDAMACS 传输时间安排算法

参 考 文 献

[1] L. Bao and J. Garcia-Luna-Aceves. Hybrid channel access scheduling in ad hoc networks. Proc. IEEE Tenth International Conference on Network Protocols (ICNP), November 2002.

[2] L. Bao and J. J. Garcia-Luna-Aceves. A new approach to channel access scheduling for ad hoc networks. In The seventh annual international conference on Mobile computing and networking 2001, pages 210-221, 2001.

[3] I. Chlamtac and A. Farago. Making transmission schedules immune to topology changes in multi-hop packet radio networks. IEEE/ACM Transactions on Networking, 2(1):23-29, February 1994.

- [4] L. Kleirock and F. Tobagi. Packet switching in radio channels, part 1: Carrier sense multiple-access models and their throughput-delay characteristics. *IEEE Transactions on Communications*, 23(12):1400-1416.
- [5] L. Kleirock and F. Tobagi. Packet switching in radio channels, part 2: Hidden-terminal problem in carrier sense multiple access and the busy-tone solution. *IEEE Transactions on Communications*, 23(12):1417-1433, 1975.
- [6] S. Ramanathan. A unified framework and algorithm for channel assignment in wireless networks. *Wireless Networks*, 5(2):81-94, 1999.
- [7] S. Ramanathan. A unified framework and algorithms for (T/F/C)DMA channel assignment in wireless networks. In *IEEE INFOCOM*, pages 900-907, 1997.
- [8] V. Rajendran, K. Obraczka, and J.J. Garcia-Luna-Aceves. Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks. *Proc. ACM Conf. Embedded Networked Sensor Systems*, Nov. 2003.
- [9] J. Grnkvist. Assignment methods for spatial reuse TDMA. In *ACM MobiHoc '00: Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing*, pages 119-124, Piscataway, NJ, USA, 2000. IEEE Press.
- [10] T. Herman and S. Tixeuil. A distributed TDMA slot assignment algorithm for wireless sensor networks. In *Proceedings of the First Workshop on Algorithmic Aspects of Wireless Sensor Networks(AlgoSensors'2004)*, number 3121 in *Lecture Notes in Computer Science*, pages 45-58, Turku, Finland, July 2004. Springer-Verlag.
- [11] T. Moscibroda and R. Wattenhofer. Coloring Unstructured Radio Networks. In *17th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, Las Vegas, Nevada, USA, July 2005.
- [12] S. Parthasarathy and R. Gandhi. Distributed algorithms for coloring and domination in wireless ad hoc networks. In *FSTTCS*, pages 447-459, 2004.
- [13] Injong Rhee, Ajit Warrier, Jeongki Min and Lisong Xu. DRAND: Distributed Randomized TDMA Scheduling For Wireless Ad-hoc Networks. In *ACM MobiHoc '06: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, Florence, Italy, 2006. ACM Press.
- [14] R. Rozovsky and P. R. Kumar. SEEDEx: a MAC protocol for ad hoc networks. In *ACM MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, pages 67-75, New York, NY, USA, 2001. ACM Press.
- [15] S. Coleri and P. Varaiya. PEDAMACS: Power Efficient and Delay Aware Medium Access Protocol for Sensor Networks. *IEEE TRANSACTIONS ON MOBILE COMPUTING*, VOL.5, NO.7, pp.920-930, July 2006.
- [16] S. Coleri. PEDAMACS: Power Efficient and Delay Aware Medium Access Protocol for Sensor Networks. technical report, Dept. of Electrical Eng. and Computer Science, Univ. of California, Berkeley, Dec.2002.
- [17] R. Ramaswami and K.K. Parhi. Distributed Scheduling of Broadcasts in a Radio Network. *IEEE INFOCOM Conf.*, vol.2, pp.497-504, 1999.

- [18] J.H. Chang and L. Tassiulas. Maximum Lifetime Routing in Wireless Sensor Networks. IEEE/ACM Trans. Networking, vol.12, no.4, Aug. 2004.
- [19] S. Coleri, M. Ergen, and T.J. John Koo. Lifetime Analysis of a Sensor Network with Hybrid Automata Modelling. Proc. ACM Int'l Workshop Wireless Sensor Networks and Applications, Sept.2002.
- [20] Using TOSSIM Simulator to Develop TinyOS Components. www.tinyos.net/tinyos-1.x/doc/tutorial/lesson5.html, May 2006.

第 4 章 无线传感器网络混合类MAC协议

4.1 斑马MAC协议 (Z-MAC)

一个常用无线网络 MAC 协议就是载波侦听多址访问 (Carrier Sense Multiple Access, CSMA) 协议。因为 CSMA 简单、灵活、强壮, 所以 CSMA 非常流行。CSMA 不需要过多的基础设施支持: 不需要时钟同步, 不需要全网拓扑信息, 能够适度处理节点动态入网和动态退网而不需要额外的操作。但是, 这些优点的代价是尝试和错误: 尝试会引起在相同时刻发送的两个或者两个以上节点产生访问碰撞, 从而导致在目的节点发生信号逼真度下降。在一个节点的任意两跳相邻区域内可能发生碰撞。尽管采用 CSMA 能够大幅度减少一跳相邻区域内发生的碰撞, 但是 CSMA 在一跳范围之外却不能发挥作用。这个问题 (一跳范围之外的碰撞问题) 称为隐含终端 (Hidden Terminal) 问题, 会导致吞吐量严重下降, 特别是在高数据传输速率的传感器应用中尤其如此。尽管 RTS/CTS 能够减轻隐含终端问题, 但是 RTS/CTS 开销高 (占 WSN 信道容量的 40%~75%), 这是因为 WSN 中的数据分组通常非常小。

TDMA 协议能够安排相邻节点的发送时间, 使相邻节点在不同时间发送, 所以能够解决隐含终端问题, 不会增加额外的消息开销。但是, TDMA 存在许多缺点。第一, 按照一种可扩展方式寻找一个高效时间安排不是一个简单任务。这经常需要一个中心节点来寻找一个无碰撞的传输时间安排。而且, 开发一个具有并发性强、信道复用程度高的高效传输时间安排非常困难 (最佳解决方法就是难以解决的)。第二, TDMA 需要时钟同步。尽管时钟同步是许多传感器应用所必需的, 但是严格时钟同步需要频繁交换信息, 所以能耗高。第三, 由于时变信道条件变化、物理环境变化、电池能量耗尽以及节点失效, 所以 WSN 可能会频繁遭遇拓扑变化。TDMA 处理动态拓扑变化代价高, 可能需要全网的变化情况。第四, 由于无线干扰范围不同于通信范围, 有些干扰节点可能没有直接处在通信范围内 (这种现象称为干扰不规则), 所以很难确定相邻节点之间的干扰关系。因此, 采用通信范围而不是采用干扰范围建立冲突关系的任何信道分配肯定得不到无干扰的传输时间安排。况且干扰范围以及信道状况时变性强, 因此一个固定传输时间安排很可能不足以保证一直不发生碰撞。第五, 在轻度竞争期间, TDMA 的信道利用率比 CSMA 低得多, TDMA 的时延高于 CSMA, 这是因为在 TDMA 协议中, 节点只能在其所分得的时隙中发送, 而在 CSMA 协议中, 只要没有竞争, 节点可以在任何时候发送。

TDMA 的这些缺点暗示一个优良的 TDMA 协议是不切实际的。即使能够得到一个高效传输时间安排, 但是诸如干扰不规则、时变信道状况、时钟同步错误之类的其他因素也会弱化 TDMA 的优点。但是, Z-MAC 协议的研究人员断定: 一个高效传输时间安排提供的信息, 特别是一个能够同时发送的独立节点集提供的信息, 可以用来减少碰撞的发生, 特别是在激烈竞争条件下碰撞的发生。

美国北卡罗来纳州州立大学开发了一个 WSN 混合 MAC 协议, 这个协议叫做斑马-MAC

(Zebra MAC, Z-MAC)。Z-MAC 协议既综合了 TDMA 和 CSMA 的优点，又弥补了 TDMA 和 CSMA 的缺点。Z-MAC 协议的主要特点是其网络竞争程度的自适应能力：在轻度竞争条件下表现类似 CSMA，在激烈竞争条件下表现类似 TDMA。Z-MAC 协议抗 WSN 中常见的动态拓扑变化和同步失败的能力强。

下面开始简单描绘 Z-MAC 协议采用的 WSN 时间同步协议 (Timing-sync Protocol for Sensor Networks, TPSN)，然后接着描述 Z-MAC 协议。

4.1.1 时间同步协议 (TPSN)

N 个传感器节点分散在区域 A 中。每个节点维护一个 16 bit 的寄存器，作为晶体振荡器触发时钟。这只是表示节点具有时间概念。TPSN 的目标是提供遵循“永动”模型的时间同步，为每个 WSN 节点建立一个公共时间尺度，同步每个传感器节点的 16 bit 时钟。

1. 基本概念

TPSN 的第一步是创建分层网络拓扑。给每个节点分配一个层次号，确保 i 层的一个节点至少能够与 $i-1$ 层的一个节点通信。在 0 层只分配一个节点，将该节点称为“树根节点”。将 TPSN 的这个阶段称为“层次号寻找阶段”。一旦完成分层结构的建立，树根节点初始化 TPSN 第二阶段（即同步阶段）。在同步阶段， i 层上的节点同步到 $i-1$ 层上的节点。最终每个节点同步到树根节点上，完成全网的时间同步。

一般情况下，作为 WSN 与其外部的网关节点的用户节点可以作为树根节点。用户节点若是配置了 GPS 接收机，则传感器节点同步到自然世界上。在比较敌对的环境中，WSN 不可能有外接实体，所以各个传感器节点可以采用某种主节点选择算法周期性轮流承担树根节点职责。TPSN 和“永动”模型均不能限制网络中存在多个树根节点的可能性。若存在多个树根节点，则在网络中构建时间同步节点孤岛。还可以采用诸如 RBS 之类的技术来维护孤岛边界上的相邻节点间的相对时钟，从而实现全网时间同步。

2. 假设条件

假设每个传感器节点具有唯一的身份识别码。链路层协议确保每个节点知道其能够进行直接通信的节点子集（称为该节点的相邻节点集）。尽管网络中可能存在单向链，但是 TPSN 只使用双向链路来实现一个节点子集中两两之间的时间同步。只使用双向链有可能构建生成树。

创建和维护网络分层结构是 TPSN 的职责。因为许多 WSN 应用要求网内处理，也需要分层结构，所以创建和维护分层结构的开销不是 TPSN 特有的开销。

3. 层次号寻找阶段

在开始（即布置 WSN）时进入层次号寻找阶段。树根节点分得层号 0 后，广播一个 level_discovery（层次号寻找）分组，启动层次号寻找阶段。level_discovery 分组包含发送节点的身份识别码和所在层次号。树根节点的直接相邻节点接收到该 level_discovery 分组后，给自己分配一个层号：比接收到的层次号大 1，即层次号为 1；设置好自己的层次号后，广播

一个新的 `level_discovery` 分组，该分组包含自己的身份识别码和所在层次号。这个过程依此继续进行下去，最终每个节点分得一个层次号。一个节点分得一个层次号后不再接收 `level_discovery` 分组，确保在层次号寻找阶段不会发生泛洪拥塞和出现 `level_discovery` 分组暴。因此，只会针对一个节点（即 0 层上的树根节点）创建分层结构。节点可能由于 MAC 层碰撞而接收不到 `level_discovery` 分组，稍后介绍这种情况的处理方法。这里使用简单泛洪机制来创建分层结构，当然也可以使用较为精确的最小生成树算法来创建分层结构，在实际中需要对精确性与复杂性进行综合平衡。

4. 同步阶段

沿着所建立起的分层结构的边沿进行节点对时间同步。采用发送节点-接收节点的传统同步方法进行节点对之间的握手。

首先分析节点对交换握手消息能够实现时间同步。图 4-1 (a) 表示节点对 A 和 B 之间交换的握手消息。 T_1 、 T_4 表示根据 A 本地时钟测得的时间， T_2 、 T_3 表示根据 B 本地时钟测得的时间。A 在 T_1 给 B 发送一个 `synchronization_pulse`（同步脉冲）分组。`synchronization_pulse` 分组包含 T_1 时间值和 A 所在层次号。B 在 T_2 接收到该 `synchronization_pulse` 分组， $T_2 = T_1 + \Delta + d$ ， Δ 表示 A 和 B 之间的时钟漂移， d 表示分组传播时延。B 在 T_3 给 A 回送 `acknowledgement`（确认）分组。`acknowledgement` 分组包含 B 所在层次号以及 T_1 、 T_2 、 T_3 时间值。A 在 T_4 接收到该 `acknowledgement` 分组。假设时钟漂移和分组传播时延在小时间范围内不会变化，那么 A 能够计算时钟漂移和分组传播时延如下：

$$\Delta = ((T_2 - T_1) - (T_4 - T_3)) / 2; \quad d = ((T_2 - T_1) + (T_4 - T_3)) / 2 \tag{4-1}$$

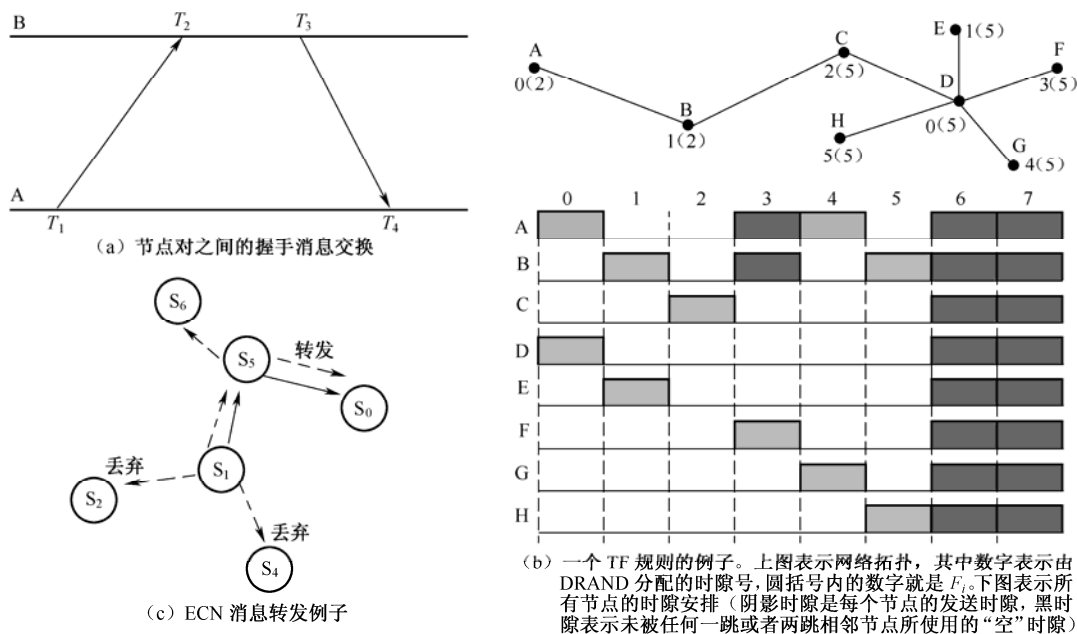


图 4-1 Z-MAC 协议说明示意图

A 已知时钟漂移 Δ ，就能够纠正其时钟，将其时钟同步到 B 上。这就是发送节点初始化同步法，即发送节点将其时钟同步到接收节点上。

上述网络级消息交换是从树根节点开始的。树根节点广播一个 `time_sync` (时间同步) 分组, 启动同步阶段。1 层上的节点接收到该 `time_sync` 分组后, 等待一段随机时间, 然后启动与树根节点的双方消息交换。随机等待时间用于避免信道访问的竞争。1 层节点接收到树根节点回送的 `acknowledgement` 分组后, 调整其时钟, 将其时钟同步到树根节点上。2 层节点旁听 1 层节点的消息交换, 这是因为每个 2 层节点至少有一个 1 层节点是其相邻节点。2 层节点旁听到 1 层节点交换消息, 退避一段随机时间, 然后启动与 1 层节点间的消息交换。这个随机退避时间用于确保 2 层节点在 1 层节点完成其时间同步后启动同步阶段。节点接收到 `synchronization_pulse` 分组而回送 `acknowledgement` 分组, 证明自身已经完成时间同步, 从而确保网络中不会出现多级同步。

在整个网络中进行这个时间同步过程, 最终每个节点同步到树根节点上。在 WSN 中经常发生分组碰撞。节点发送完 `synchronization_pulse` 分组后等待 `acknowledgement` 分组, 经过一段随机时间发生超时, 却仍然没有接收到 `acknowledgement` 分组, 则重发 `synchronization_pulse` 分组。继续进行这个过程, 直到双方成功完成消息交换为止。

5. 其他有关问题的处理

在 WSN 中, 经常采用随机方式布置节点, 存在这种情况: 一个传感器节点加入一个已经建立的网络, 即加入层次号寻找阶段已经结束的网络。即使节点在网络开始时存在, 但是由于 MAC 层碰撞, 该节点可能接收不到 `level_discovery` 分组。在任何一种情况下, 该节点得不到分层号。但是, 每个节点必须是分层结构的组成部分, 才能够与树根节点同步。因此, 当布置一个节点的时候, 该节点等待一段时间, 以便得分层号; 若在等待时间内没有得分层号, 则发生超时, 然后该节点广播一个 `level_request` (分层号请求) 分组。相邻节点应答时回送自己的层次号。该节点给自己分配一个层次号: 该层次号比已经接收到的最小层次号大 1。该节点完成加入该分层结构。

传感器节点可能随机失效。存在这种情况: 一个 i 层节点没有 $i-1$ 层相邻节点。此时, 这个 i 层节点无法接收到其发送的 `synchronization_pulse` 分组的 `acknowledgement` 分组, 因此不能同步到树根节点上。为了处理碰撞问题, 节点等待一段随机时间后仍然没有接收到 `acknowledgement` 分组, 重发 `synchronization_pulse` 分组。节点重发若干次 (重发次数恒定) 后仍然没有接收到 `acknowledgement` 分组, 则假定已经丢失所有上一层相邻节点, 于是广播一个 `level_request` 分组。该节点接收到 `acknowledgement` 分组后, 分得一个新的分层号。假定网络仍然连通, 那么该节点至少有一个相邻节点, 因此确保在分层结构中分得一个新分层号。推荐重发次数为 4, 用于确定上层某个相邻节点无效。重发次数大于 4, 则所需同步时间增大; 重发次数小于 4, 则会出现不必要的泛洪, 降低同步精度。

假如树根节点失效, 那么 1 层节点接收不到的 `acknowledgement` 分组, 此时出现等待超时, 采用上述方法进行处理。但是, 1 层节点不是广播 `level_request` 分组, 而是执行某种主节点选择算法, 选出的主节点作为新的树根节点。新树根节点重新开始同步过程, 重新启动层次号寻找阶段。

4.1.2 Z-MAC协议概述

Z-MAC 协议使用 CSMA 作为基准 MAC 协议, 但是使用 TDMA 传输时间安排作为提高

竞争解析度的“启示”。在 Z-MAC 协议中，在布置 WSN 的时候进行时隙分配，所以在开始时开销较高。T-MAC 协议的设计原理是起始高开销，通过一段长时间的网路工作初步得到补偿，最终通过吞吐量和能量效率的提高得到补偿。在 Z-MAC 协议中使用 DRAND 算法。DRAND 是一个高效可扩展信道传输时间安排算法，是 RAND 算法的分布式实现。RAND 是一个集中式信道复用传输时间安排算法。完成时隙分配后，每个节点在每个预先确定的周期（称为“帧”）中周期性重复使用所分得的时隙。将所分得时隙的那个节点称为该时隙的占有节点，其他节点称为该时隙的非占有节点。由于 DRAND 允许两跳相邻区域外的任意两个节点拥有同一个时隙，所以每个时隙可能有多个占有节点。

与 TDMA 不同的是，节点可以在 Z-MAC 协议的任意时隙发送。节点在一个时隙发送之前（在该时隙开始时不是必要的），总是进行载波侦听的，当载波侦听信道空闲时发送一个分组。但是，该时隙的一个占有节点在该时隙访问信道的优先权高于其非占有节点。调整初始竞争窗口大小，使每个占有节点的发送总是先于非占有节点，从而实现信道访问的优先权。其目的是，占有节点在其所占时隙上有数据需要发送时，其发送优先于非占有节点，所占时隙具有碰撞回避优先权，因此 Z-MAC 协议减少了碰撞机会；但是当一个时隙未被其占有节点使用时，非占有节点可以借用这个时隙。这种优先权方案自然会影响到根据竞争程度而进行 CSMA 和 TDMA 之间的切换。这种优先权方案的一个重要特性是可调整占有节点的信道访问概率，而非占有节点无关。这个特性对提高 Z-MAC 协议抗同步失败和时隙分配失败能力、同时提高其对竞争的可扩展性具有重要作用。

通过将 CSMA 和 TDMA 综合在一起，Z-MAC 协议的抗定时失败、时变信道状况变化、时隙分配失败以及拓扑变化能力强于单独的 TDMA；在最坏情况下，Z-MAC 协议总是退回到 CSMA。由于 Z-MAC 协议只要求两跳相邻区域内发送节点同步，所以 Z-MAC 协议研究人员设计了一个简单的本地同步算法，即每个发送节点根据其当前数据传输速率和资源预算调整其同步频率。分析表明：即使在时钟完全不同步以及存在一定程度时隙分配失败条件下，Z-MAC 协议的性能仍然与 CSMA 相当。

Z-MAC 协议包括一个建立过程，在建立过程中依次执行如下操作：相邻节点寻找、时隙分配、本地帧交换、全网时间同步。这些操作只是在建立过程中执行一次，不会执行到网络拓扑发生重大变化（比如重新布置传感器）的时候。其思想是执行这些操作的初始开销通过数据传输期间吞吐量和能量效率的提高来补偿。

下面首先描述这些操作，然后讨论将这些操作与 Z-MAC 协议的主要传输控制机制综合在一起。

4.1.3 相邻节点寻找与时隙分配

一个节点开始启动时，首先执行一个简单相邻节点寻找协议。该协议周期性地给其一跳相邻节点广播一条 ping 消息，将其一跳相邻节点信息收集在一张列表中。ping 消息包含一张其当前一跳相邻节点列表。在实验中，每个节点每秒随机发送一条 ping 消息，连续 30 s。在这个过程中，每个节点从 ping 消息中收集其一跳相邻节点的信息，这样必然组成其两跳相邻节点的信息。

两跳相邻节点列表作为时隙分配算法的输入。Z-MAC 协议采用 DRAND 算法给网络中每个节点分配时隙，确保所建立的广播传输时间安排不存在两跳相邻区域内任意两个节点分

得同一个时隙的问题。这种时隙分配保证一个节点对其任意一个一跳相邻节点的发送不会受到其两跳相邻节点任意发送的干扰。注意：广播传输时间安排能够处理一跳相邻节点间的任意路由变化问题。

DRAND 的时隙分配与网络大小无关，但是与每个节点的本地相邻区域大小有关，所以 DRAND 具有可扩展性能。DRAND 建立效率极高的传输时间安排，分配给节点的时隙号不会大于本地两跳相邻区域大小 $O(\delta)$ ——在大多数情形下是前者小于后者。DRAND 的运行时间和消息复杂性也受 $O(\delta)$ 限制。可见，DRAND 的能耗与本地相邻区域大小呈线性正比关系。在少量新节点入网的时候，DRAND 能够进行本地时隙分配，但是不会改变已经分配给现有节点的时隙分配。DRAND 详情请参阅第 3 章。

4.1.4 本地成帧

完成时隙分配后，要求每个节点立即确定使用其时隙的发送周期。每个节点的发送周期叫做该节点的时间帧（Time Frame, TF）。传统处理方法：所有节点必须保持一个相同时间帧，同时所有节点相互同步，其时隙 0 同步在相同时刻上。这种处理方法要求将最大时隙号（Maximum Slot Number, MSN）传播到整个网络中，不需要适应本地时隙变化。在网络增加新节点的时候，DRAND 能够进行本地时隙分配，同时维持现有时隙分配。假如这种分配方法导致 MSN 变化，那么必须将变化后的 MSN 传播到整个网络中。这就可能导致以高开销适应网络中小范围变化。在 Z-MAC 协议中，无线信道状况不稳定引起的网络拓扑变化由 Z-MAC 协议自然处理，所以不会引起新的时隙分配，但是新节点入网或者节点重新布置则可能引起新的时隙分配。

下面介绍一种新方法，每个节点利用这种方法来维护适合其本地相邻区域大小的本地时间帧，避免与其他竞争节点发生任何冲突。

1. 时间帧（TF）规则

设节点 i 由 DRAND 分得一个时隙 s_i ，节点 i 的两跳相邻区域内的 MSN 为 F_i 。设节点 i 的时间帧为 2^a ，其中选择的正整数“ a ”满足条件 $2^{a-1} \leq F_i < 2^a - 1$ 。也就是说，节点 i 在每个 2^a 时间帧中使用第 s_i 个时隙（对于所有 $l=1,2,3,\dots$ ，时隙为 $l2^a + s_i$ ）。

定理 4.1 假如每个节点 i 只使用时隙 $l2^a + s_i$ ， $l=1,2,3,\dots$ ，那么节点 i 的两跳相邻区域内不存在使用节点 i 所使用时隙的节点 j 。

TF 规则允许节点根据其本地两跳信息选择自己的时间帧，使 DRAND 自适应时间帧的动态变化（由本地拓扑变化引起），而不会引起全网变化。图 4-1（b）给出一个采用 TF 规则得到的 TDMA 传输时间安排的例子。假如采用一个全网时间帧，那么时间帧长度为 6。因此，尽管节点 A 和 B 的帧长均为 2，但是节点 A 和 B 每隔 6 个时隙只能使用一次自己的时隙。但是假如使用 TF 规则，那么就允许节点 A 和 B 的帧长均为 4。这就提高了信道使用的同时性，减小了节点 A 和 B 的消息时延。但是，从图 4-1（b）中看到时隙 6 和 7 未分配给相邻区域内的任何节点。这是一种平衡：当网络均匀密集时，全网时间帧产生的空闲时隙较少。但是假如网络包含许多稀疏区域，只有少数几个密集区域，那么最好采用本地成帧。在 Z-MAC 协议中，空闲时隙对 CSMA 是可用时隙，所以空闲时隙不会被浪费。

2. 同步到时隙 0 上

本地成帧规则假定每个节点在相同时刻从其时隙 0 开始。假定时钟同步，并且预先将时隙 0 同步在一个绝对时间值上，那么就很容易实现且不需要任何通信开销。例如，可以将开始真实时间（即同步时钟值为 0）设为时隙 0 的开始时间。新节点的时钟同步到全网时钟上，因而新节点很容易同步其时隙。为了实现这种同步，Z-MAC 协议只是在开始进行一次全网时钟同步。完成初始同步后，每个节点执行低开销本地同步协议。

4.1.5 Z-MAC协议的传输控制

DRAND 过程结束后，每个节点将其帧长和时隙号转发给自己的两跳相邻区域。因此，节点在 Z-MAC 阶段开始时就知道其一跳和两跳相邻节点的有关时隙信息和帧信息。此时，每个节点同步到时隙 0 上，然后就完全准备好执行 Z-MAC 协议的传输控制规程。

在 Z-MAC 协议中，节点有两种工作方式：轻度竞争（Low Contention Level, LCL）方式和激烈竞争方式（High Contention Level, HCL）。一个节点只有在最近一个 t_{ECN} 周期内接收到其一个两跳相邻节点发送的一条直接竞争通知（Explicit Contention Notification, ECN）消息后才按照 HCL 方式工作。否则，节点按照 LCL 方式工作。节点正在激烈竞争时就发送一条 ECN 消息。

在 LCL 工作方式下，任意节点可以在任意时隙完成发送；但是在 HCL 工作方式下，只有当前时隙的占有节点以及其一跳相邻节点才被允许竞争信道的访问权。在两种工作方式中，时隙占有节点的优先权高于非占有节点。假如一个时隙没有占有节点或者其占有节点在该时隙没有数据发送，那么非占有节点可以借用这个时隙。即使在轻度竞争条件下，这种特性也能够实现高信道利用率，这是因为节点能够实现有多少可用信道就能够发送多少信息。Z-MAC 协议采用 B-MAC 协议的退避、CCA、LPL 接口实现 LCL 和 HCL。

4.1.6 发送规则

节点 i 有数据需要发送时，首先检查自己是否为当前时隙的占有节点。节点 i 若是占有节点，则在一个固定时间周期 T_0 内进行随机退避。完成随机退避后，节点 i 执行 CCA：假如信道空闲，则节点 i 发送数据；假如信道不空闲，则节点 i 等待信道不忙时再进行上述过程。假如节点 i 不是当前时隙的占有节点并且处在 LCL 工作方式，或者假如节点 i 处在 HCL 工作方式并且当前时隙未被其两跳相邻节点所占用，那么节点 i 等待 T_0 ，然后在一个竞争时间窗口 $[T_0, T_{no}]$ 内进行随机退避。完成随机退避后，节点 i 执行 CCA：假如信道空闲，则节点 i 发送数据；假如信道不空闲，则节点 i 等待信道不忙时再进行上述过程。假如节点 i 不是当前时隙的占有节点并且处在 HCL 工作方式（这意味着节点 i 的一个两跳相邻节点在最近 t_{ECN} 内发送了一条 ECN 消息），那么节点 i 推迟其发送（可以休眠），一直推迟到找到一个时隙为止，该时隙的占有节点不是其两跳相邻节点，或者该时隙的占有节点就是节点 i 。节点 i 苏醒后重复上述过程。

根据上述发送规则，虽然各个时隙的优先级不同，但是节点 i 在 LCL 工作方式时能够竞争任何时隙。在 HCL 工作方式时，如果当前时隙的占有节点是节点 i 或者是节点 i 的某一个

跳相邻节点,那么节点*i*能够竞争当前时隙。注意:在前一个时隙开始的发送延长到一个 HCL 时隙,有可能与该 HCL 时隙的占有节点发生碰撞。防止这个问题的一种方法是限制发送不能延长到 HCL 时隙。这样做可能会产生过多分片,导致信道利用率下降。因为时隙足够长,可供处理一个以上的分组,所以 HCL 时隙给占有节点更多成功机会,而不会发生隐含终端问题。

T_o 和 T_{no} 的取值影响性能。根据最大有效吞吐量的随机分析选择 T_o 和 T_{no} 的取值, T_o 取值的选择决定 Z-MAC 协议面对时间同步错误和时隙分配失败问题的强壮性,时隙分配失败导致有些时隙具有多个占有节点。假如时间同步错误不大于一个 TDMA 时隙长度,那么任何时候最多存在 2~3 个相互冲突的占有节点。据此将 T_o 设为 8 个竞争窗口时隙(即对于有效实现等于 2 的幂)。在 IEEE 802.11 中, T_{no} 通常作为初始竞争窗口的大小,为此将 T_{no} 设为 32 个时隙。

以前 Ephremides 和 Mowafi 采用概率 TDMA (Probabilistic TDMA, PTDMA) 方法,针对无线局域网(或者一跳)环境研究过根据竞争程度在 TDMA 和 CSMA 之间无缝自适应 MAC 协议。正如 TDMA 那样,实际时间被时隙化,通过调整时隙占有节点的访问概率(a)和非占有节点的访问概率(b),PTDMA 根据竞争程度在 TDMA 和 CSMA 之间进行自适应 MAC 操作。通过函数 $a+(M-1)b=1$ 来调整访问概率 a 和 b ,这里 M 表示发送节点数量。尽管 PTDMA 和 Z-MAC 协议的设计目标相同,但是 PTDMA 主要是为一跳无线局域网环境设计的,没有处理 TDMA 在 Ad Hoc WSN 中面临的许多问题,比如时间同步错误、干扰无规律、拓扑变化等问题。这些问题可能引起 PTDMA 性能急剧下降。PTDMA 假定发送节点具有缓存能力,所有节点经历相同的统计到达时间。在一个只有一个节点子集是活动数据源(WSN 中很常见)的网络中,PTDMA 表现出极差的信道利用率,性能表现不如 CSMA,这是因为:访问概率 a 和 b 相互依赖, a 不减小(减小 a 导致 PTDMA 表现不如 TDMA),则 b 不能任意增大。访问概率 a 的影响也不清楚。作者似乎希望针对不同竞争程度调整访问概率 a ,但是文中没有提到如何实现。Z-MAC 协议不需要通过动态调整其参数来达到所需要的效果。

Z-MAC 协议的发送规则不同于 PTDMA。与 PTDMA 不同的是,Z-MAC 协议占有节点和非占有节点的信道访问概率[式 $a+(M-1)b=1$ 中的 a 和 b]由 T_o 和 T_{no} 来单独调整,因为非占有节点不能在 T_o 期间竞争信道访问权。这就提高了 Z-MAC 协议的强壮性,且不会影响 Z-MAC 协议的一般性能。例如,增大 T_o 不会改变占有节点和非占有节点的优先级,从而使 Z-MAC 协议的性能依据竞争程度在 TDMA 和 CSMA 之间波动。在 PTDMA 中,这是不可能的,因为依赖 a 和 b 。

4.1.7 直接竞争通知

直接竞争通知(Explicit Contention Notification, ECN)消息用来通知两跳相邻节点在激烈竞争时不要成为每个时隙占有节点的隐含终端。每个节点根据其本地竞争程度的估计就地决定 ECN 消息的发送。有两种方法估计两跳相邻节点的竞争程度。一种方法是接收一跳相邻节点的应答,测量应答分组的丢失率。由于两跳相邻节点竞争引起碰撞,所以这种方法与应答分组丢失率密切相关。但是这种方法要求接收节点发送反馈信息,产生额外开销。除非应用要求应答,否则应答开销会导致信道利用率过度下降,消耗能量,缩短网络寿命。另外一种方法是测量信道的噪声等级。竞争越激烈,噪声等级越高。因为可以在传输数据之时被动测量噪声等级,所以这种方法不会产生任何额外开销。为了被动测量噪声等级而不用主动采

样信道，测量发送节点在发送之前消耗的平均噪声退避量。一个噪声退避量等于发送节点在发送分组（发送节点只是在信道干净之时才发送）之前采用 CCA 侦听信道时所花费的退避时间。若噪声等级高于 CCA 门限值，则发送节点进行退避。为了理解噪声退避与两跳相邻节点竞争之间的相关性，做一个 Mica2 实验：两个分群向一个公共接收节点（称为中心节点）发送，相同分群内的节点相互之间相距一跳远，不同分群的节点相互之间相距两跳远，在一个分群内固定一个发送节点（这个节点叫做测试节点），而在另一个分群内发送节点数量可变，测量中心节点处两跳相邻节点竞争与测试节点处噪声等级之间相关性随着发送节点数及其发送速率变化的变化情况。按照中心节点每秒经历如下情况的次数测量两跳相邻节点竞争程度：中心节点退出空闲状态进入接收状态，但是由于数据被破坏或者采样数据中噪声高（包括失步、CRC 校验出错以及前导失败）而未接收到数据。

在低速发送时，即使增加发送节点，发现平均噪声等级仍然低于每个分组 0.1 个噪声退避量，两跳相邻节点竞争程度仍然低于 5 次每秒。但是，将发送速率增大到全速（所有发送节点总是有数据发送）后，平均噪声等级却仍然位于每个分组 0.2 个噪声退避量之上。

发送节点检测到激烈竞争情况时，给正参与竞争的那个节点发送一条单目标一跳 ECN 消息。假如存在多个竞争节点，那么发送节点发送一条广播消息，广播消息包含有关多个竞争节点的信息。在 WSN 中，通常由于每个节点有一个父节点向自己发送，所以一个节点有一个目的节点。节点 j 接收到其一跳相邻节点 i 触发的一条一跳 ECN 消息时，首先检查自己是否为该条消息的目的节点：假如是，那么节点 j 给其一跳相邻节点广播一条 ECN 消息（这些 ECN 消息称为两跳 ECN 消息）；假如节点 j 不是该消息的目的节点，那么节点 j 丢掉该条 ECN 消息。一个节点接收到一条两跳 ECN 消息后，将其 HCL 标志置位。图 4-1 (d) 给出一个 ECN 消息转发例子。节点 S_1 遇到严重分组丢失情况时，给节点 S_5 发送一条一跳 ECN 消息。实箭头表示 ECN 消息转发路径，虚箭头表示 ECN 消息。节点 S_2 和 S_4 不是目的节点，所以将该 ECN 消息丢掉。节点 S_5 将该 ECN 消息广播给自己的一跳相邻节点。由于 HCL 被激活，所以节点 S_6 和 S_0 不会在节点 S_1 所占有的时隙上竞争信道，而节点 S_5 、 S_2 、 S_4 可以作为一跳相邻节点以低于节点 S_1 的概率竞争信道。

HCL 标志只是一个软状态，其含义如下：除非在最近 t_{ECN} 周期内接收到另外一条两跳 ECN 消息，否则 HCL 标志将被复位。因此，假如节点 i 连续经历竞争，那么节点 i 必须周期性地发送 ECN 消息。ECN 消息的刷新周期 t_{ECN} 由系统设置。

通常，在一个节点检测竞争情况之时，很可能其相邻发送节点也在同时检测竞争情况。所转发的 ECN 消息中存在重复多余的 ECN 消息。为了防止出现 ECN 消息暴，采用旁听来抑制 ECN 消息。节点 i 检测到激烈竞争情况时，在发送下一条一跳 ECN 消息之前先进行随机退避。同时，节点 i 若是接收到一条发送给另一个节点的一跳 ECN 消息，并且这个节点的目的节点与自己的目的节点相同，则抑制自己的 ECN 消息，取消发送这条 ECN 消息。经过 t_{ECN} 时间后，假如节点 i 仍然经历激烈竞争，那么节点 i 采用随机退避，安排另一条 ECN 消息的发送时间，重复上述过程。对路由转发节点同样采取 ECN 消息抑制规则。路由转发节点在 t_{ECN} 周期内接收到一条一跳 ECN 消息并且已经转发了一条 ECN 消息，则不转发两跳 ECN 消息。

ECN 消息类似于 CSMA/CA 中的 RTS/CTS 控制分组。但是两者的区别在于 HCL 使用拓扑信息（即时隙信息）来避免两跳碰撞。ECN 消息只是在激烈竞争时才会被触发，所以 ECN 消息的开销比 RTS/CTS 低得多。采用 ECN 消息抑制技术后，只有少量 ECN 消息才需要被转

发。由于 HCL 状态持续时间可能大于单个分组的传输时间，所以 ECN 开销被分摊在多个分组传输上。也可以将 ECN 消息看做类似于 CODA 中的抑制消息，两者的区别在于：ECN 只抑制 ECN 消息产生节点所占有时隙的两跳相邻节点，而抑制消息抑制所有接收节点但消息中指定的接收节点除外。

4.1.8 Z-MAC 传输时间安排的接收

DRAND 只定义了节点的传输时间安排，而在 Z-MAC 协议中，节点可以在任何时隙发送。Z-MAC 协议是在 B-MAC 协议上面实现的，所以作为默认配置使用 LPL，每个节点维护侦听占空因数，占空因数之间的间隔为检查周期，每次发送之前先发送前导，前导长度等于检查周期。因此，Z-MAC 协议的空闲侦听能耗，特别是在低占空因数条件下，是可以与 B-MAC 协议比拟的。

检查周期也是确定时隙大小的一个因素，这是因为一个时隙必须足够大，可供发送一个分组。因此，一个时隙长度必须大于检查周期 T_o 、 T_{no} 、CCA 周期以及一个分组传播时间之和。时隙大小和网络时延之间需要综合平衡，特别是在激烈竞争下。在轻度竞争下，节点可以在任何时候发送，所以时隙大小不会影响时延；但是在激烈竞争下，节点处在 HCL 方式，并且只能在少数几个分配好的时隙上发送。因此，时隙长，则时延大。把时隙大小的选择任务交给应用来完成。应用设计者必须评估时隙大小和网络时延之间的综合平衡，找出符合要求的时隙长度。

4.1.9 本地时间同步

采用载波侦听和拥塞退避技术后，Z-MAC 协议具有强于 TDMA 的抗时钟错误能力。在没有同步的条件下，Z-MAC 协议的性能退回到 CSMA 的性能。即使在轻度竞争条件下，Z-MAC 协议也能像 CSMA 那样工作，可有时钟同步，也可没有时钟同步。因此，Z-MAC 协议在激烈竞争条件下需要时钟同步来实现 HCL。Z-MAC 协议的一个重要特征就是只是在处于激烈竞争中的相邻发送节点之间才需要时钟同步。由于只是本地相邻发送节点之间才需要同步，所以这些特点提供一种极好的机会，优化时钟同步开销，并且可以根据发送节点的发送速率调整同步频率，发送速率较高的发送节点的同步消息发送频率也较高。在这种同步方案中，接收节点被动地将其时钟同步到发送节点时钟上，不需要发送任何同步消息。

为了实现发送节点之间的本地时钟同步，Z-MAC 协议采用了 RTP/RTCP（实时传输协议）中的技术。在 RTP/RTCP 协议中，限制控制消息传输速率，使其只占会晤带宽的一小部分，每个会晤成员根据所分得的会晤带宽调整其控制消息的发送速率。在 Z-MAC 协议中，每个发送节点将其同步消息所占用带宽限制在其数据发送速率的预先确定的一小部分 (B_{synch}) 内（如每隔 100 个数据分组发送一个同步分组）。事实上，每个节点可以独自运用其某种能量和带宽预算函数确定其同步消息所占带宽。在 Z-MAC 协议中将 B_{synch} 设为发送速率的 1%。

在这个本地时钟同步协议中，每个数据发送节点周期性地发送同步消息，同步消息包含发送节点的当前时钟值。一个节点接收到一条同步消息后，使用其当前时钟值和新收时钟值的加权移动平均值更新其时钟值。由于只有发送节点发送同步消息，所以轻流量区中的有些节点的时钟值可能漂移，与其他同步节点的时钟值偏移大。当这些节点开始发送的时候，其

时钟未同步（Mica2 的最大时钟漂移速率约为 40 μ s）。因此，这些节点的时钟是不可信时钟。为了避免参照未同步发送节点的时钟，采用一个信任因子（ β_i ）来调整平均权重。 β_i 反映同步消息发送节点的同步频率，通过所发送和接收的同步消息的频率来计算。

设每个传感器的最大时钟漂移速率为 r_{drift} ，最大可接受时钟错误为 $\varepsilon_{\text{clock}}$ 。那么 $I_{\text{synch}} = \varepsilon_{\text{clock}} / r_{\text{drift}}$ 决定达到最大时钟错误或者更小时钟错误所要求的最小同步周期。设 S 表示一个节点发送或者接收同步消息的平均速率， α_{synch} 表示所收新时钟值的最大权。那么可以计算出该节点的 β_i ： $\beta_i = \min\{\alpha, S \times I_{\text{synch}} \times \alpha_{\text{synch}}\}$ 。对新收时钟值 C_{new} 和 C_{avg} 加权移动取平均，计算出一个时钟的加权移动平均值 C_{avg} ： $C_{\text{avg}} = (1 - \beta_i)C_{\text{avg}} + \beta_i C_{\text{new}}$ 。

在 Mica2 中，节点为了在每秒 40 μ s 的最大漂移速率下维持 1 ms 时钟精度以及维持每 100 个分组（分组大小为 49 B）一个同步分组的同步周期，那么所维持的数据发送速率和（或者）数据接收速率必须大于或者等于 1.5 kb/s。在 1.5 kb/s 传输速率下，节点的信任因子变为 α_{synch} ，同步只占发送速率的 1%，即 150 b/s（或者每秒 1/3 个分组，分组大小为 49 B）。假如数据传输速率（发送速率和接收速率之和）低于 1.5 kb/s，那么节点的信任因子可忽略不计。但是，这不会对吞吐量造成不利影响，这是因为在 1.5 kb/s 以下的低数据传输速率下，节点很可能不会遇到过多竞争，并且 CSMA 在轻度竞争条件下有效工作。

在上述同步方案中，发送和接收同步消息的节点常常具有较大的信任因子，在更新时钟值中对信任因子的取值有较深的反映。通常情况下，路由路径上的节点往往比其他节点发送更多的分组，因而具有较大的信任因子。类似地，不常发送数据的源节点具有较小的信任因子。当一个源节点经过一段长时间休眠后重新开始发送数据时，其时钟可能已经漂移，与其他较好同步的时钟偏移较大。但是随着源节点速率的增大以及其数据不断朝中心节点传递，其时钟值越来越接近其他路由转发节点的时钟值。在实验中发现，即使 30 个节点孤立而没有同步，这 30 个孤立节点仍然能够在 10 条同步消息内与网络中其他节点重新同步。

4.1.10 Z-MAC协议的性能

1. 实验方法

在 Mica2/TinyOS 测试床中实现 Z-MAC 协议，评估 Z-MAC 协议性能。

除非特别说明，否则采用 B-MAC 协议的默认设置（请参阅第 2 章）。由于 Z-MAC 协议是在 B-MAC 上面实现的，所以采取的分组格式与 B-MAC 相同（见第 2 章中的表 2-4）。B-MAC 协议的默认初始退避窗口大小和拥塞退避窗口大小分别为 32 个时隙和 16 个时隙（每个时隙 400 μ s）。除了吞吐量实验需要改变退避窗口大小来观测退避窗口大小对信道利用率的影响之外，其他实验全部采用默认退避窗口大小。Z-MAC 协议参数的默认值如表 4-1 所示。

表 4-1 Z-MAC 协议参数的默认设置

TinyOS 和 ns-2 的实验参数	默 认 值
占有节点竞争窗口大小 (T_o)	8 个时隙
非占有节点竞争窗口大小 (T_{no})	32 个时隙
竞争窗口每个时隙的长度	400 μ s
ECN 刷新周期 (t_{ECN})	10 s

续表

TinyOS 和 ns-2 的实验参数	默认值
时间同步平均权重 (α_{synch})	0.25
最大时钟漂移速率 (r_{drift})	40 μs
最大时钟错误 (ϵ_{clock})	1 ms
同步带宽 (B_{synch})	1%
Z-MAC 的 TDMA 时隙长度	50 ms
通信距离 (ns-2)	200 英尺
干扰距离 (ns-2)	300 英尺
通信带宽 (TinyOS, ns-2)	19.2 kb/s

对于真实的多跳方案，建立一个 42 个 Mica2 节点的网络，每个节点位于美国北卡罗来纳州立大学（NCSU）计算机科学大楼的系办公室和教室中。图 4-2（a）表示测试床和节点间的无线通信链。在这个测试床中，所有节点的最大两跳相邻区域为 27，本地最大帧长 32（很多节点的帧长小于 32）。为了排除路由差异的影响，在所有实验中采用固定路由路径。图 4-2（b）表示测试床中 30 个节点使用的三条路由路径。节点 36 为中心节点。较粗的线表示承载较重流量的链路。

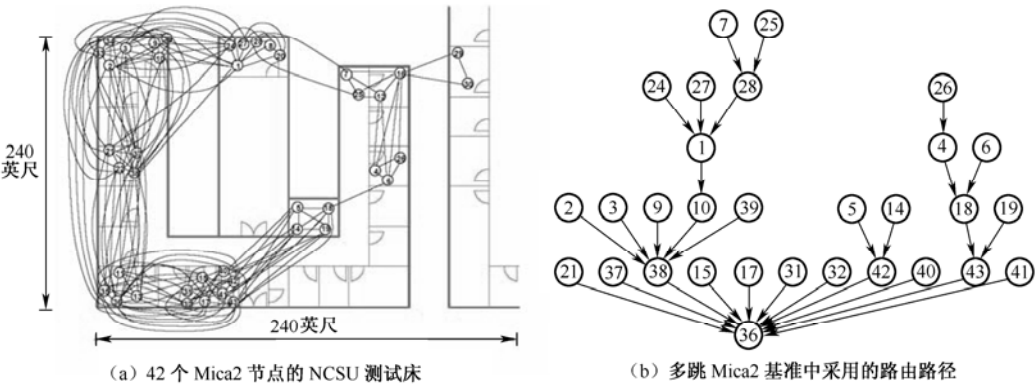


图 4-2 Z-MAC 协议实验配置

2. 吞吐量

在吞吐量仿真实验中，测试和比较 B-MAC、Z-MAC 两个 MAC 协议的有效信道利用率。只按照所做过的 B-MAC 协议数据吞吐量实验（见第 2 章）测试数据吞吐量：每个分组为 36 B，其中数据为 29 B，分组头为 5 B，CRC 校验为 2 B，测试每个分组的有效数据载荷。

图 4-3（a）给出了 Z-MAC-HCL 和 B-MAC 两个协议在 42 个 Mica2 节点多跳基准下的数据吞吐量实验结果曲线。传输速率低于 3.12 个分组/秒时，Z-MAC-HCL 和 B-MAC 两个协议交付全部分组，达到大致相同的吞吐量。B-MAC 协议的数据吞吐量稍优于 Z-MAC 协议。这是因为 Z-MAC 协议非占有节点的退避拥塞窗口 ($T_o+T_{no}=8+32=40$) 大于 B-MAC 协议的退避拥塞窗口 (16)。因为竞争程度轻，Z-MAC 协议的大多数发送都是按照非占有节点来进行的，所以退避时间造成 Z-MAC-HCL 和 B-MAC 两个协议的数据吞吐量稍有不同。随着传输速率

大于 3 个分组/秒后，从图 4-3 中观察到 Z-MAC 协议的数据吞吐量比 B-MAC 协议高出约 20%~30%。在满传输速率（50 个分组/秒）下，Z-MAC 协议达到约 7.2 kb/s 的吞吐量，而 B-MAC 协议达到约 5.2 kb/s 的吞吐量。这些数据结果稍高于轻度竞争下 Mica2 节点二跳基准下的实验结果。这是因为网络非常密集，所以各个节点能够非常容易地相互侦听到对方，所以一跳相邻节点竞争对两跳相邻节点竞争起主导作用。

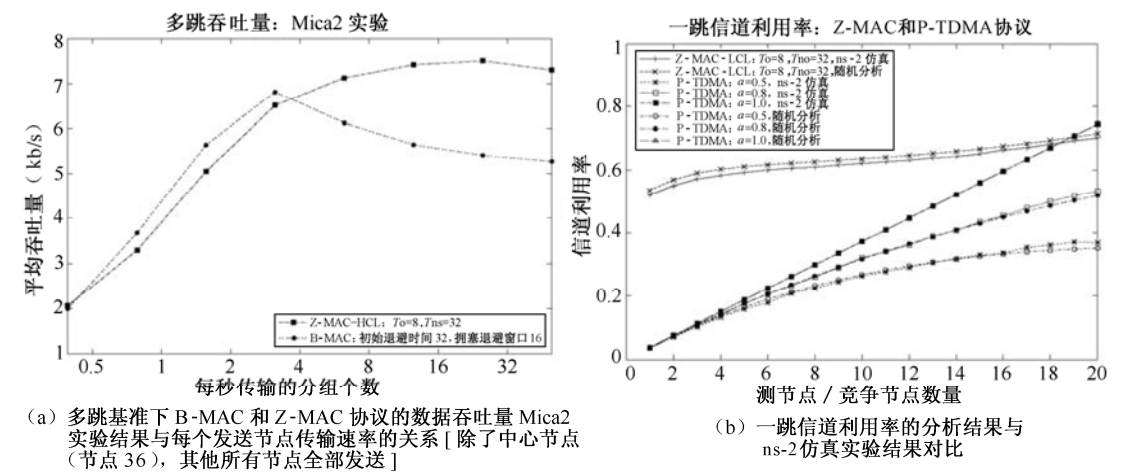


图 4-3 Z-MAC 协议的性能

3. 能量效率

表 4-2 给出了 Z-MAC 协议在多跳基准下建立阶段操作的详细能耗。重复运行 30 次 Z-MAC 协议建立阶段，报告平均能耗和标准偏移。在建立阶段平均每个节点 7.22 J 的能耗，占每个节点（2 500 mAh、3 V 电池，表 2-3 也采用这种电池）总有效能量的 0.03%。尽管 DRAND 协议和其他操作没有进行节能优化，但是对比每次发送的能耗，这仍然是相当重要的一部分能耗。其思想是，开始时的能耗通过随后 Z-MAC 协议规律性发送期间的能量效率的提高来补偿。下面概括 Mica2 基准下的能量效率实验结果。

表 4-2 多跳 Mica2 测试床中 Z-MAC 协议建立阶段操作期间的平均能耗（单位：J）

操 作	平均能耗/J	平均偏移量
相邻节点寻找	0.73	0.0018
DRAND	4.88	3.105
本地帧交换	1.33	1.39
TPSN	0.28	0.036

4.1.11 Z-MAC协议随机分析

1. P-TDMA协议随机分析

考虑一个 TDMA 时隙 i 的占有节点是 O_i 。 O_i 或者处于活动状态或者处于非活动状态。

O_i 若是一个活动源节点，则以概率 a 在 i 中发送，剩余 $B-1$ 个活动源节点以概率 b 在 i 中发送（ B 表示网络中活动源节点数量）。 O_i 若是非活动源节点，则 B 个活动源节点全部以概率 b 在 i 中发送。设一个活动源节点在一个时隙中成功发送一个分组的概率为 P_{s_a} ，一个非活动源节点在一个时隙中成功发送一个分组的概率 $P_{s_{in}}$ 。

$$P_{s_a} = a(1-b)^{B-1} + b(1-a)(B-1)(1-b)^{B-2} \quad (4-2)$$

$$P_{s_{in}} = Bb(1-b)^{B-1} \quad (4-3)$$

式中， P_{s_a} 表示时隙占有节点赢取其时隙，同时所有其他 $B-1$ 个非占有源节点未获取该时隙的概率，或者表示时隙占有节点未获取该时隙、一个非占有源节点赢取该时隙、剩余 $B-2$ 个非占有源节点未获取该时隙的概率（因为可能有 $B-1$ 个这种赢取节点，所以需要乘以 $B-1$ ）。 $P_{s_{in}}$ 表示一个源节点以概率 b 赢取时隙所有其他 $B-1$ 个源节点未获取该时隙的概率（因为可能有 B 个这种赢取节点，所以需要乘以 B ）。已知这些概率，则可以计算信道利用率 $S(N,a,b)$ 为

$$S(N,a,b) = \{[P_{s_a} \times B + P_{s_{in}} \times (N - B)] / N\} \times (T_d / T_p) \quad (4-4)$$

对活动时隙和非活动时隙中达到的信道利用率加权求和，得到平均信道利用率。因子 T_d/T_p 表示分组头所占带宽比率。

2. Z-MAC协议随机分析

考虑一个 TDMA 时隙 i 的占有节点是 O_i 。 O_i 在 i 中进行随机退避发送，退避时间窗口为 T_0 ；而时隙 i 的非占有节点等待 T_0 ，然后在 i 中进行随机退避发送，退避时间窗口为 T_{no} 。显然， O_i 总是具有较高优先权在其占有时隙中发送。假如占有节点没有数据发送，那么非占有节点就可以相互竞争，竞争时间窗口为 T_{no} 。

设时隙占有节点在固定窗口 T_0 内选择一个随机退避时间，非占有节点在固定窗口 T_{no} 内选择一个随机退避时间，占有节点的平均窗口大小 $W_0=(T_0-1)/2$ ，非占有节点的平均窗口大小 $W_{no}=((T_0-1)+(T_{no}-1))/2$ 。假定节点间严格时间同步，那么占有节点由于其退避窗口较小而每次都能赢取信道。因此，在 N 个时隙的 B 个时隙中，相应的赢取节点总是能够获取时隙。在刚好一个竞争节点的情况下，Z-MAC 协议表现类似时隙化、无记忆的 CSMA。因此选择时隙 r 的概率受到均匀分布的控制，即

$$P_s(r) = 1/W \quad (4-5)$$

设占有节点获得的信道利用率表示为 $S_0(1,W_0)$ 。在剩余的 $N-B$ 个时隙中，所有 B 个源节点在固定窗口 W_{no} 内相互竞争。在 B 个竞争节点情况下，Z-MAC 协议表现类似时隙化、固定窗口、无记忆的 CSMA。因此得到

$$P_s(r) = 1/W_n \quad (4-6)$$

在 $N-B$ 个时隙中获得的信道利用率表示为 $S_{no}(B,W_{no})$ 。已知 S_0 、 S_{no} ，则平均信道利用率 $S(N,B,T_0,T_{no})$ 等于 B 、 $N-B$ 个时隙的信道利用率的加权平均，即

$$S(N,B,T_0,T_{no}) = S_0(1,W_0)(B/N) + S_{no}(B,W_{no})(1-B/N) \quad (4-7)$$

3. 性能分析对比

将 P-TDMA 的每个 TDMA 时隙长度设为 19.1667 ms，足够发送一个完整分组；而将

Z-MAC 的每个时隙长度设为 50 ms。图 4-3 (a) 同时给出了 P-TDMA、Z-MAC 的随机分析结果与 ns-2 仿真实验结果。

当 P-TDMA 越来越类似 CSMA 而不是越来越类似 TDMA 的时候,特别是当 $a=0.5$ 的时候, P-TDMA 的性能非常接近随机分析结果。当 $B \ll N$ 时, P-TDMA 性能恶化,原因如下:设计的发送概率 a 和 b 只允许每个节点在 N 个时隙内发送一个分组,因此,当网络中只有少数几个活动源节点时,得到极低的信道利用率。提高信道利用率的一种直观方法是使 a 和 b 较为适应信道竞争的情况:理想结果是,在低竞争程度时 P-TDMA 表现类似 CSMA,而在高竞争程度时 P-TDMA 表现类似 TDMA。这正是 Z-MAC 协议的操作方法。

从图 4-3 (b) 中看到,显然 Z-MAC 协议对所有竞争程度下的信道利用进行了有效管理。低竞争程度下的信道利用率稍低点,这是因为几乎没有活动源节点发送,常常是非占有节点发送,因此存在 $T_o + \text{random}(T_{no})$ 退避时延。随着竞争程度的提高, Z-MAC 协议的信道利用率越来越接近 $a=1$ 时的 P-TDMA (纯 TDMA) 信道利用率。Z-MAC 协议的随机分析结果稍高于 ns-2 仿真实验结果,这是因为在 Z-MAC 协议中,常常是分组长度频繁超越时隙边界,而上述分析模型假定所有分组都在时隙边界 (一个时隙) 内,不存在重叠。

4.2 漏斗-MAC协议

WSN 表现出独特的漏斗效应,按照多点对一点流量模式,传感器场中产生的事件朝一个或者多个中心节点方向逐跳传递,最终到达中心节点,如图 4-4 (a) 所示。逐跳通信与中心节点数据集中收集相结合,产生事件自由流出 WSN 的阻塞点。例如,事件漏斗导致事件不断朝中心节点传递而不断接近中心节点,传输流量和传输时延不断增大,从而引起严重的分组碰撞、拥塞、分组丢失,其最好结果就是中心节点有限的应用逼真度,最差结果就是 WSN 拥塞崩溃。事件漏斗还存在其他缺点。离中心节点最近 (通常相距几个转发跳) 的传感器节点丢失更多分组,丢失数量不均匀,将这个漏斗区域称为强度区,如图 4-4 (a) 所示。强度区内的传感器节点消耗大量能量,并且多于离中心节点较远 (强度区之外) 的传感器节点,从而导致整个网络寿命的缩短。减轻漏斗效应是 WSN 研究人员和团体面临的一个重要挑战。

研究人员已经提出了分布式拥塞控制算法[比如合成拥塞控制技术 (FUSION)、拥塞检测与预防 (CODA) 等]、网络分层设计技术、数据累积技术来处理 WSN 中载荷不断增加、拥塞不断加重的问题。但是其中单独任何一项技术都非常难以有效地控制聚集点或者源节点的流量速率,使其匹配中心节点的瓶颈条件,因而很难完全减轻漏斗效应问题。WSN 中主要分组丢失发生在离中心节点最近几跳范围内,即使在轻流量条件下也是如此。在离中心节点最近几跳范围内或者离中心节点更远范围内采取新的控制,能够获得通信性能的重大改善,根除漏斗效应。

美国哥伦比亚大学的研究人员提出和设计了一个本地化的、面向中心节点的漏斗-MAC 协议 (Funneling-MAC)。在漏斗-MAC 协议设计中明确认可存在漏斗效应。漏斗-MAC 协议是 TDMA (安排类) 和 CSMA/CA (竞争类) 两种 MAC 协议的混合协议,用于事件漏斗的强度区内,如图 4-4 (a) 所示。纯 CSMA/CA 是漏斗-MAC 协议的一个组成部分,作用于整个网络。漏斗-MAC 协议通过只在强度区内采用本地 TDMA 传输时间安排来减轻漏斗效应,对越接近中心节点的节点提供机会越高的传输时间安排,越接近中心节点的节点承载的流量比远离中心节点的节点多得越多。漏斗-MAC 协议将强度区内传感器事件的 TDMA 传输时间

管理任务安排由中心节点来承担和完成，因而是面向中心节点的 MAC 协议。由于只是在接近中心节点的强度区内而没有在整个传感器场中采用 TDMA，所以漏斗-MAC 协议是一个本地化的 MAC 协议。强度区深度的计算和维护也由中心节点来承担和完成。假定中心节点很可能比简单传感器节点有较强的计算能力和较多的储能，但是漏斗-MAC 协议的有效操作并不依赖这一点。通过本地化方法采用 TDMA，以及将较多的管理任务交由中心节点而不是传感器节点来承担和完成，提供一种可扩展解决方法在 WSN 中采用 TDMA，这种解决方法能够提升中心节点的应用逼真度，但是不存在全网使用 TDMA 的有关扩展性问题。

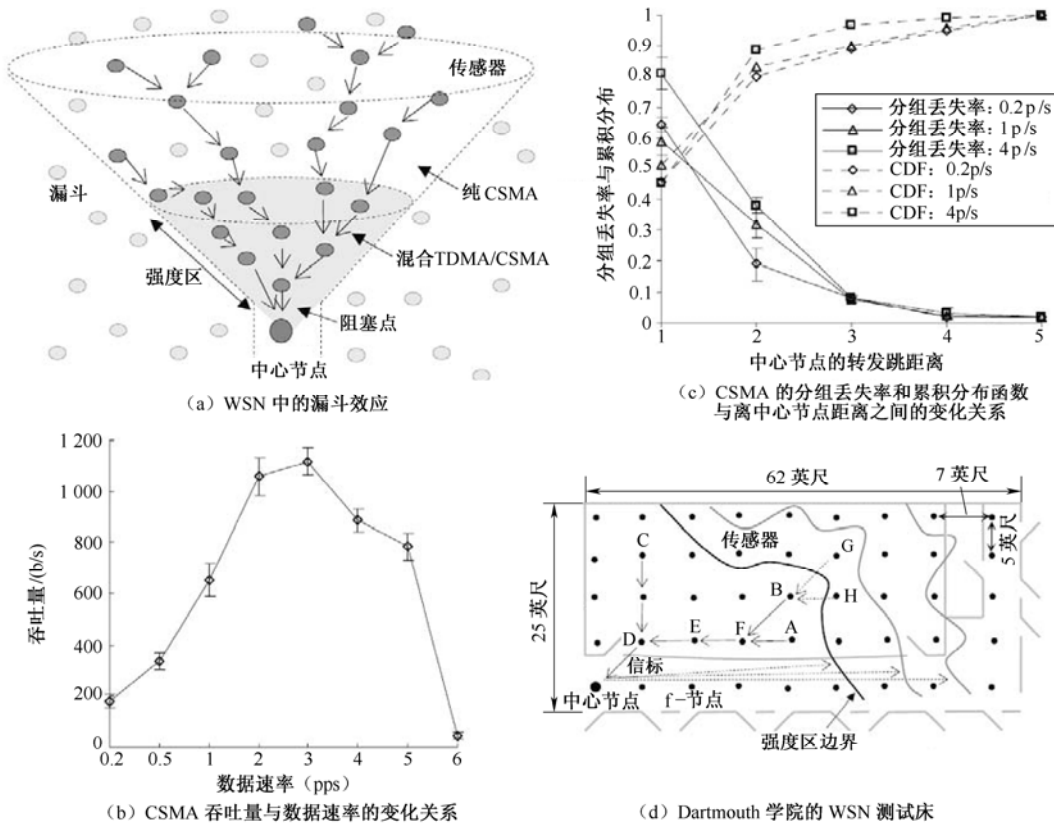


图 4-4 漏斗问题

4.2.1 漏斗问题

首先定量说明一个 WSN 中漏斗效应的影响，这个 WSN 是由 45 个 Mica2 节点组成的网络测试床，使用 TinyOS 基于 CSMA 的 B-MAC 协议、MintRoute 路由协议、Surge 应用。网络布置在一个大房间内，45 个 Mica2 节点等距离布置在一个 5×9 的矩形栅格内，确保实验中不会相互干扰以及出现近场效应 (Near-Field Effect)，如图 4-4 (d) 所示，其中左下角的 Mica2 节点为中心节点。所设置的节点间隔和发射功率满足一跳相邻节点达到 80% 以上的分组交付率、两跳相邻节点达到 20% 以下的分组交付率。从而为实验建立一个相当严格而又密集的多跳无线通信环境。

从 44 个传感器节点 (中心节点除外) 中随机选出 16 个节点来产生事件 (作为源节点)，

产生速度为 0.2~5 个分组/秒 (Packets Per Second, PPS), 分组长 36 B。实验目的是让 WSN 逐步承载由轻量到中等的流量, 然后进入拥塞和饱和状态, 研究、测试中心节点阻塞点吞吐量以及网络中的分组丢失情况。消息通过多跳传递, 在 Dartmouth 学院 WSN 测试床实验中是 2~5 跳, 最终到达中心节点。图 4-4 (b) 表示在中心节点处测得的逼真度 (即吞吐量曲线) 与 16 个发送节点数据速率之间的变化关系。注意在计算吞吐量逼真度时没有考虑前导和 CRC, 而将一个分组计算为 36 B。从图 4-4 (b) 中可以明显看到: 中心节点吞吐量在网络进入拥塞和饱和状态前出现约 1 100 b/s 的峰值。进一步增大事件产生速率只会进一步导致网络过载, 并且网络最终随着载荷的提高而崩溃。从图 4-4 (b) 中可以看到: 轻流量事件产生速率为 0.2 pps, 中等流量 (接近最佳载荷) 事件产生速率为 1 pps, 超载流量事件产生速率为 4 pps。使用这些事件产生速率来进一步研究网络中的分组丢失分布情况。考虑网络总分组丢失率等于网络中丢失分组数量与网络中发送分组数量之比。在事件产生速率为 0.2 pps、1 pps、4 pps 下分别测得的总分组丢失率分别为 67%、72%、95%。这个测试结果令人吃惊, 在轻流量下仍然存在相当高的分组丢失率 (67%), 而在重流量下分组丢失率竟高达 95%。高分组丢失率转化为能量的大量浪费。这么高的分组丢失率将很快耗尽传感器的储能, 所以很多应用不能接受这么高的分组丢失率。注意: 在轻流量和中等流量下, 分组丢失的主要原因是碰撞和隐含终端问题; 而在重流量和超载流量下, 分组丢失的原因除了碰撞和隐含终端问题外, 还包括缓存器溢出。

下面考虑分组丢失率在多跳网络传输中的分布问题。图 4-4 (c) 中的实线表示离中心节点 l 跳处的分组丢失率 (第 l 个转发跳节点发送和丢失的分组数量与第 l 个转发跳节点发送的分组数量之比)。结果明显定量地说明了实验中的漏斗效应, 说明漏斗效应引起了网络性能下降、变差。图中结果是 5 次运行同一个实验的平均结果, 置信区间达 95%。重要结果是, 图 4-4 (c) 明显表明了离中心节点越近, 分组丢失率越高, 这是漏斗效应多点到一点逐跳传递流量模式的结果。例如, 对于各种流量速率, 大量分组丢失发生在离中心节点一跳至二跳处, 进一步远离中心节点 (两跳以上), 分组丢失率迅速下降。这就是漏斗效应的特点。即使是 0.2 pps 的轻流量载荷, 离中心节点最近几跳的分组丢失仍然在所有分组丢失中起主导作用。低速流量的这些每跳分组丢失率特性说明了在 0.2 pps 低速流量下网络分组总丢失率高达 67% 的原因。图 4-4 (c) 中的虚线表示每跳分组丢失率的累积分布函数 (Cumulative Distribution Function, CDF)。从图 4-4 (c) 中 CDF 曲线可以看到: 在低速、中速、高速流量下, 离中心节点最近两跳范围内的分组丢失率在 80%~90% 之间。所以由此得到结论: 漏斗效应几乎不随事件源节点产生速率变化而变化。

这些结果指出: 对于实验中所考虑的全部流量速率 (即低速、中速、高速), 在离中心节点最近几跳范围内施加新的控制 (比如传输时间安排) 能够获得性能的重大提高。可以得出结论: 即使在低速流量下, 基于 CSMA 的 B-MAC 协议也不能减轻漏斗效应。这些都是重要的领悟和认识。因此推断需要优于 B-MAC 的新 MAC 协议来彻底解决漏斗问题。

下面将详细描述漏斗-MAC 协议。顾名思义, 漏斗-MAC 协议是解决前述漏斗问题的 MAC 协议。

4.2.2 按需发送信标

中心节点广播的信标触发漏斗-MAC 本地化 TDMA。所有传感器节点在默认方式下运行

CSMA，若接收到信标则认为自己是 **f**-节点。中心节点通过控制信标的发射功率来调整强度区的边界，如图 4-4（d）所示。通过动态深度调整算法来确定信标发射功率。中心节点按照计算出来的发射功率发送信标。接收到信标的传感器节点认为自己处于强度区内，并且是 **f**-节点，然后运行 **TDMA**；而没有接收到信标的传感器节点（比如强度区外面的传感器节点）运行 **CSMA**。

f-节点必须时钟同步才能够运行 **TDMA**，但是漏斗-**MAC** 协议不需要任何时钟同步协议。假如网络中存在同步协议，那么漏斗-**MAC** 协议也能够使用该同步协议，进一步减少其活动信标信令。但是在测试床上实现漏斗-**MAC** 协议时假定网络中不存在同步协议，而是在信标消息中嵌入一个微型时钟同步协议。因此，**f**-节点依靠所发送的信标来激活 **TDMA**、调整强度区边界，实现时钟同步。一个传感器节点只要接收到一个信标，就成为 **f**-节点，初始化其时钟，实现其时钟与其他 **f**-节点同步。**WSN** 的信标传播时延为微秒级，而漏斗-**MAC** 协议要求的时钟同步精度为毫秒级，所以采取基于信标的同步技术能够维持足够精度的时钟同步，做出 **TDMA** 传输时间安排。因为信标是整个强度区的广播信标，所以所有 **f**-节点在相同时刻接收信标，因而同步紧凑。

信标分组包含少数几个控制域：信标发送间隔时间（周期）、超帧持续时间、**TDMA** 持续时间。超帧持续时间和 **TDMA** 持续时间稍后解释。根据经验设置信标发送间隔时间，以便响应可能发生的路由变化、流量速率变化以及 **f**-节点时钟漂移。根据传感器节点本地时钟精度以及同步精度要求确定信标发送间隔时间。

只是在必要之时和按需基础上发送信标，在网络空闲或者正在接收极低速流量时不发送信标。每个 **f**-节点维持一个定时器，假如在一段大于信标发送间隔时间的时间内没有接收到信标，则定时器定期满。节点在其定时器期满后运行纯 **CSMA** 协议。中心节点按照流量加权移动平均的变化，只要从全部路径上接收到足够多的数据分组，就按照计算出来的信标发送间隔时间周期性发送信标。反之，中心节点若是在一个或者多个信标发送间隔时间内没有接收到网络时隙分配所需要的足够多的数据分组，则停止发送信标，直到中心节点记录这种变化情况为止。**f**-节点利用信标发送间隔时间同步中心节点随后的信标发送。采取基于传感器节点的信标发送间隔时间定时器，假如传感器节点本次信标发送可能干扰另一个信标，则推迟这次信标发送。

中心节点在网络启动时或者在空闲周期结束后开始发送信标，开始采用最低发射功率（普通传感器节点的发射功率）发送。这是因为深度调整算法使用递增式增大/减小规则来计算信标/传输时间安排的发射功率。随着所测流量的增大，中心节点逐步增大发射功率，采用动态深度调整算法满足吞吐量/分组丢失率要求。反之，中心节点在网络启动时或者在空闲周期结束后开始以最大发射功率发送信标，那么所发送的信标很可能干扰 **CSMA** 输入数据分组。这是因为传感器节点在启动状态下或者正好在空闲周期结束后不知道何时发送信标。这个问题由漏斗-**MAC** 协议来解决，因为动态深度调整算法的起始发射功率总是相同于传感器节点使用的公共默认功率（这被认为是深度调整算法的功率平面）。因此，干扰的影响被降到最低程度。因为深度调整算法的目标就是提高强度区的深度，因此发射功率的问题就是，当深度调整算法增大信标发射功率时，采用现有发射功率不可达节点将会被干扰。漏斗-**MAC** 协议通过引入“传输时间安排广告元”来解决这种干扰问题，

漏斗-**MAC** 协议的设计目标是采用按需技术限制周期性广播信标的开销。另外考虑的一个参数是增大信标发送间隔时间，平衡信令开销及现有强度区内传感器节点使用的接收功率，

降低中心节点的能量需求。引入“惰性信标”概念，由此提出用于维护 f-节点紧凑时钟同步和传输时间安排的信标发送间隔时间。假如按照这种方式不受限制地推出信标发送间隔时间，那么可能会引起一定的性能下降。稍后讨论用于维护紧凑时钟同步和传输时间安排以及最佳吞吐量的最佳信标发送间隔时间，并与惰性信标对比，惰性信标允许将信标发送间隔时间设为最佳信标发送间隔时间的 3 倍，结果只是网络性能稍有下降。

4.2.3 面向中心节点的传输时间安排

中心节点监视其每条聚集路径的输入流量，根据所监视的流量（最初只是根据新 CSMA 事件，随后包括现有 TDMA 流量）计算全部路径的 TDMA 传输时间安排，然后按照信标发送功率广播传输时间安排分组而实现传输时间安排的分发。

将聚集路径定义为多条路径（两条或者两条以上）在进入强度区时或者之前合并而成的一条路径。漏斗-MAC 协议将一条聚集路径作为一条单独路径条目来处理。如在图 4-4（d）中，漏斗-MAC 协议按照一条单独聚集路径条目 B—F—E—D 保存有关路径 G—B—F—E—D 和 H—B—F—E—D 的信息。由于进入强度区的聚集路径数量受到强度区内节点数量的限制，所以漏斗-MAC 协议具有良好的扩展性。下面将详细讨论面向中心节点的传输时间安排问题。图 4-5 给出了漏斗-MAC 算法的伪码。

为了计算传输时间安排，中心节点需要确定各条聚集路径的路径头 f-节点的身份以及该聚集路径上的加权平均流量，这样才能够正确安排该聚集路径的传输时间。聚集路径概念表示事件序列从聚集路径头节点[如图 4-4(d)中的传感器节点 A]开始，沿着由 TinyOS MintRoute 路由协议确定的一条路由[即图 4-4（d）中的聚集路径 A—F—E—D—中心节点]，朝中心节点方向，逐跳向前传递，最终到达中心节点。中心节点测试每条聚集路径流量的加权移动平均值，根据时隙分配规则分配时隙。为了使中心节点能够获得聚集路径流量信息，漏斗-MAC 协议在分组头中预留 3 B，这 3 B 构成一个聚集路径信息域。聚集路径信息域只由强度区内特定聚集路径上的 f-节点来更新，包含一个路径头节点身份（ID，2 B）子域和转发跳数子域。中心节点从每条聚集路径的输入分组中收集这些信息，用于安排强度区内全部聚集路径的传输时间。聚集路径头节点位于强度区边界附近，聚集路径头节点 ID 等于聚集路径头节点的节点身份，转发跳数表示聚集路径头节点与中心节点之间聚集路径的转发跳数传输距离。例如在图 4-4（d）中，若节点 A 接收到强度区外面产生的一个事件分组，那么节点 A 朝中心节点方向沿着聚集路径 A—F—E—D—中心节点转发这个事件分组。在这个实验中，聚集路径头节点 ID 是节点 A，转发跳数等于 4。重要的是，节点 A 接收到一个聚集路径信息域为零的数据事件分组时，将自己作为聚集路径头节点。此外，强度区内部源节点产生一个新分组时，也将自己作为聚集路径头节点。聚集路径头节点将聚集路径头节点 ID 子域设为自己的身份（ID），将转发跳数子域设为 1。聚集路径上的每个 f-节点在转发这个事件数据分组时将转发跳数子域的值加 1。因此，传递到达中心节点的每个事件数据分组携带有其传递通过聚集路径的头节点和转发跳数。

中心节点监视其输入数据分组，持续跟踪每条聚集路径的输入流量速率及其聚集路径头节点 ID、转发跳数（见图 4-5 中“中心节点流量测试”伪码）。中心节点按照每条路径方式将每条聚集路径的流量速率保存在聚集路径表中。采样周期等于一个超帧时间，中心节点按照每条聚集路径一个超帧测试输入分组数量，然后计算每条聚集路径所测得流量的加权移动平均值。

中心节点按照每条聚集路径而不是按照每个节点计算、分配传输时间安排，如图 4-5 中“中心节点计算传输时间安排”伪码。这是因为中心节点只有有关聚集路径的信息，而没有聚集路径中有关节点的信息。这就使得漏斗-MAC 协议是可扩展的，与特定路由协议生成的路由树无关，即漏斗-MAC 协议通过简单提取聚集路径的端节点和转发跳数，而不是根据拓扑路由信息来计算传输时间安排。因此，漏斗-MAC 协议没有路由协议信息和路由树信息。中心节点将每条聚集路径的状态信息存储在聚集路径表中，采用聚集路径头节点 ID 检索聚集路径表。聚集路径表还保存有每条聚集路径的测试统计值。每个聚集路径表条目包含一个聚集路径头节点 ID、转发跳数、流量输入速率。流量输入速率表示每条聚集路径在一个超帧时间内应该承载的分组数量。中心节点每隔一个信标发送间隔时间就递减每个聚集路径表条目的有效时间；假如聚集路径表溢出，则用新条目替代最旧的条目。

```
# 中心节点流量测试
Event (Received a packet)
{
    for (path=0; path<num_path; path++){
        if (path_head_id[path] = packet -> path_head_id){
            sampled_rate[path] += 1
            num_hops[path] = packet -> num_hops
        }
    }
}
Event (End of Sampling period)
{
    for (path=0; path<num_path; path++){
        traffic_rate[path] =  $\alpha$ *traffic_rate[path]+(1- $\alpha$ )*sampled_rate[path]
        if (traffic_rate[path] > max_rate) max_rate = traffic_rate[path]
    }
}

# 传 0 传输时间 0
for (i=1; i< num_field_in_schedule_packet; i++)
{
    for (j=0; j < num_my_path_head; j++){
        if (schedule_packet_path_head_id(i)=my_path_head_id(j)){
            my_slot(slot_num + num_hops_from_path_head) = TRUE
        }
    }
    slot_num = slot_num + num_hops
}

# 中心节点计 0 传输时间 0
for (i=0, j=0; i < max_rate; i++)
{
    for (path=0; path<num_path; path++){
        if (i = 0 or traffic_rate[path] >= i){
            scheduled_slot[j] = num_hops[path]
            scheduled_path_head_id[j] = path_head_id[path]
            j = j+1
        }
    }
}
for (i=0; i < j-1; i++)
{
    if (scheduled_slot[i+1] > 3){
        scheduled_slot[i] = scheduled_slot[i] - (scheduled_slot[i+1] - 3)
        if (scheduled_slot[i] < 1) scheduled_slot[i] = 1
    }
}
total_slot = total_slot + scheduled_slot[i]
if (total_slot > max_slot) scheduled_slot[i] = 0
}

# 中心节点动态 0 调
If (beacon_power < max_power)
{
    If (total_slot < max_slot) beacon_power = beacon_power + step
    else if (beacon_power > min_power) beacon_power = beacon_power - step
}
```

图 4-5 漏斗-MAC 算法的伪码

1. 时隙分配规则

中心节点运用聚集路径表中的信息给每条聚集路径分配时隙。例如，假定一条聚集路径的流量速率为 k 、转发跳数为 h ，那么应该给该条聚集路径上的每个节点分配 k 个时隙，所以给该条聚集路径分配 $k \times h$ 个时隙。假如一条聚集路径的流量速率小于 1，那么中心节点不遵循上述规则，而是给该条聚集路径分配 $1 \times h$ 个时隙。当一个超帧内数据产生速率小于一个分组或者偶尔产生事件驱动流量时，流量速率小于 1。正如前面所述，漏斗效应即使在轻流量载荷条件下以及随着载荷的增大也仍然起作用，所以仍然需要安排流量速率小于 1 的聚集路径的传输时间。假如一条聚集路径流量速率低，那么中心节点应该给这条聚集路径分配最少的时隙。中心节点能够给一个节点分配的最少时隙数是 1。因此，中心节点应该给该聚集路径上的每个节点分配一个时隙，所以分配给该聚集路径的时隙数为 $1 \times h$ 。由这个规则可得到良好结果，这是因为测试床实验结果表明：对比 CSMA，漏斗-MAC 协议提高了轻流量条件下的网络吞吐量。

2. 简单的空间复用

为了提高漏斗区域内的吞吐量，中心节点采用空间复用。在没有完整的网络拓扑信息条件下设计最佳空间复用方案是极其困难的。但是，中心节点只使用每条聚集路径的转发跳数信息就能够计算出次佳空间复用。漏斗-MAC 协议采用这种简单次佳法，实现两个相距 3 跳或者 3 跳以上的节点复用同一个时隙。此时，f-节点使用漏斗-MAC 协议进行载波侦听，即对所安排的信道访问也进行载波侦听，以及在产生干扰时进行退避，所以不太可能产生干扰。例如，在图 4-4 (d) 中，f-节点 A 或者 B 与 f-节点 D 相距 3 跳远，共享同一个时隙。传输时间安排算法允许 f-节点 B 在 f-节点 A 的时隙之后的第 3 个时隙（即属于 f-节点 D 的时隙）开始发送。因此，计算得到的传输时间安排如下：将 3 个时隙分配给聚集路径 A—F—E—D，将 4 个时隙分配给聚集路径 B—F—E—D。

中心节点完成传输时间安排计算后，在下一个信标之后立即给其聚集路径表中记录的全部聚集路径广播一个传输时间安排分组，发射功率与信标相同，因此强度区内的所有 f-节点很可能接收到该传输时间安排分组。由于通常不会在每个信标发送间隔时间发送新的传输时间安排，所以中心节点在信标分组头中设置一个传输时间预期比特。传输时间安排分组的有效载荷包含所安排聚集路径的头节点 ID 以及分配给聚集路径的时隙数量。按照二维数组 [聚集路径头节点 ID (2 B)，时隙数量 (1 B)] 将最后得到的每条聚集路径的传输时间安排存储到传输时间安排分组的有效载荷中。如在图 4-6 (a) 所示的简单传输时间安排分组中，所有 f-节点得到通知：强度区内安排了 3 条活动聚集路径的传输时间，这 3 条聚集路径分别分得 3、4、3 个时隙。

f-节点接收到传输时间安排分组后，计算出分配给自己的那些时隙，如图 4-5 中的“传感器传输时间安排”伪码。每个 f-节点保存一张表，用于存储通过自己的每条聚集路径的头节点 ID 以及到达这个头节点的转发跳数。f-节点运用这张表就能够计算出分配给自己的时隙。例如，节点 E 维护的二维数组 [聚集路径头节点 ID，时隙数量] 条目为 [A, 2] 和 [B, 2]，所以节点 E 明白自己可以在节点 A、B 的时隙后的第二个时隙发送。

聚集路径头节点	A, 3	B, 4	C, 3
---------	------	------	------

(a) 传输时间安排分组结构

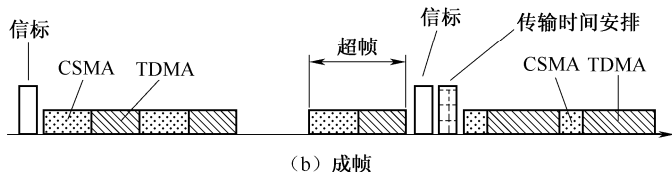


图 4-6 帧格式

4.2.4 定时与成帧

f-节点接收到一个传输时间安排分组后，将其通信同步到漏斗-MAC 帧结构上，如图 4-6 (b) 所示。f-节点在其所分得的 TDMA 时隙上发送自己安排的分组。为了提高漏斗-MAC 协议的强壮性和灵活性，在连续两个 TDMA 帧（已安排的访问周期）之间插入一个 CSMA 帧（随机访问周期），执行载波侦听，侦听所安排的发送。将 TDMA 帧和 CSMA 帧综合在一起形成一个新的帧，将其称为超帧。两个信标之间重复几个超帧，如图 4-6 (b) 所示，一个传输时间安排分组通常跟随在信标之后。

CSMA 帧的作用是在没有分到时隙时，仍然还可以安排发送的源节点发送其产生的事件数据分组。由此引起的其他问题：管理、路由、突然要求传输的新节点产生的事件数据。还有一个在测试床中经常遇到的问题：由于无线信道反复无常导致路由变化，因而在一条聚集路径上出现新的事件数据。中心节点采用其流量测试算法检测这些事件。在强度区域内提供一定程度 CSMA 访问的另外一个理由是为了支持发送异步管理与控制分组（如路由消息、hello 消息）以及重传那些在 TDMA 帧中没有成功发送的事件数据分组。重传策略只是漏斗-MAC 协议的一个可选组成部分，在链路可靠性要求下才被激活。

发送给 f-节点的信标包括 f-节点正确安排其流量传输时间和超帧 CSMA 竞争访问所必需的全部帧定时信息。从图 4-6 (b) 中看到：超帧持续时间固定不变，而 TDMA 持续时间动态变化。因为中心节点使传输时间安排适应超帧持续时间，所以超帧持续时间对性能不会产生重大影响。中心节点测试每个超帧的输入流量，根据采样过程的结果计算传输时间安排。若中心节点已采样流量速率发生变化，则 TDMA 持续时间随之变化。假如流量载荷充分递增，则中心节点在一个超帧中分配更多时隙，因而 TDMA 持续时间增大，在强度区内得到传输安排的事件更多。将超帧中未被 TDMA 使用的那部分时间分配给 CSMA 帧。在测试床实验中，将一个超帧中 TDMA/CSMA 最大比率限制为 80%，以确保最小 CSMA 分配，支持控制分组和非事先安排的数据分组的发送。

漏斗-MAC 协议提高了强壮性，执行载波侦听，侦听所安排的传输，避免可能与异常发送发生的碰撞，比如强度区内存在没有接收到信标或者 Meta 传输时间安排广告的节点，这些节点的发送可能与强度区内已安排的传输发生碰撞。最后，漏斗-MAC 协议采用 B-MAC 协议建议的低功率侦听（Low Power Listening, LPL）算法和前导技术来降低占空因数 WSN 的能耗。但是与 B-MAC 协议不同的是，f-节点是通过超帧的同步通信，不需要按照 LPL 方式发送长前导，而是采用标准短无线前导。在 TDMA 访问期间，f-节点在其所安排侦听时隙

起点苏醒；在 CSMA 访问期间，f-节点根据 B-MAC 协议建议的周期性苏醒。在 CSMA 访问期间，因为所有 f-节点能够在相同时刻苏醒和侦听信道，所以节点可以采用标准前导发送。强度区外的节点在发送数据前按照 LPL 方式采用长前导。

4.2.5 Meta-传输时间安排的广播

漏斗-MAC 协议是混合 MAC 协议，中心节点信令（即信标、传输时间安排）广播要覆盖整个强度区，发射功率可能较高，所以采用漏斗-MAC 协议会产生许多干扰问题。为了使中心节点采用高功率发送信令而不会干扰网络中正在进行的传感器通信（比如在两个传感器节点之间朝中心节点进行 CSMA 转发），节点必须能够从信标消息中获取详细的超帧定时信息。另外一个干扰问题：强度区内的某个（些）节点可能接收不到信标（比如由于信道衰落、非对称链等原因），因而没有信标中携带的定时和成帧信息，所以可能成为潜在的干扰节点。还有一种干扰：位于强度区边界外的节点不接收信标，因而没有漏斗-MAC 协议的帧定时信息，所以也是潜在的干扰节点。为了解决这些干扰问题（即分配类访问和竞争类访问之间的干扰），漏斗-MAC 协议在 f-节点接收到一个新传输时间安排后发送的第一个事件数据分组中嵌入一个低开销 Meta 传输时间安排广告。

以信标发送间隔时间为周期，接收到信标和传输时间安排的所有 f-节点在发送给中心节点的第一个事件数据分组中嵌入 Meta-传输时间安排。最小传输时间安排包含如下信息：超帧持续时间、TDMA 持续时间、当前 TDMA 帧的剩余时间，本次发送间隔时间结束前的超帧重复次数。Meta 传输时间安排仅为 4 B，承载数据事件。

强度区内没有接收到信标的节点以及强度区边界外附近的节点能够旁听到由事件数据分组承载的 Meta 传输时间安排的发送。这些节点接收到 Meta 传输时间安排后，被允许在当前超帧的 CSMA 访问期间进行发送，以便降低干扰的可能性。考虑一个例子：强度区内一条聚集路径上中间某个节点没有接收到信标。例如，在图 4-4（d）中，已安排聚集路径 A—F—E—D 的传输时间，但是节点 F 没有接收到信标。该聚集路径头节点（f-节点 A）发送一个数据分组，该数据分组包含 Meta 传输时间安排，节点 F 接收到这个数据分组。因此，节点 F 确定该数据分组是安排在当前时隙发送的，因而立即发送这个数据分组。节点 F 使用 CSMA 帧发送其另一个数据分组。假定还未安排好聚集路径 A—F—E—D 的传输时间，聚集路径头节点（f-节点 A）使用 CSMA 帧发送一个数据分组，数据分组包含其聚集路径信息域。节点 F 接收到这个包含聚集路径信息域的数据分组后，更新其转发跳数域，转发这个数据分组，所以中心节点仍然能够安排聚集路径 A—F—E—D 的传输时间。因此，Meta 传输时间安排广告能够实现强度区内 TDMA 操作与强度区外 CSMA 操作之间的无缝互操作。按照这种方法采用 Meta 传输时间安排解决了可能存在的错误操作。

4.2.6 动态深度调整

动态深度调整算法能够使漏斗-MAC 协议实现最大的中心节点吞吐量以及最小的中心节点分组丢失率。中心节点通过控制信标广播的发射功率，调整 TDMA 强度区的边界。中心节点通过确定漏斗中强度区的最佳深度 d ，能够动态改变信标的发射功率，可见 TDMA 区是积极能动的。通过分析方法来分析漏斗-MAC 协议确定最佳强度区深度的方法。下面的分析指

出：满足吞吐量最大、分组丢失率最小的最佳值 d 可以由中心节点来决定。这个结果推动了动态深度调整算法的设计。

1. 强度区最佳深度分析

下面分析强度区的最佳深度。强度区达到最佳深度时，吞吐量最大，分组丢失率最低。动态深度调整算法能够调整强度区深度，使其达到最佳深度。

因为 WSN 的 MAC 协议将吞吐量作为其主要性能指标，所以分析时采用的主要性能指标是最大吞吐量。

参考文献[20]对无线网络吞吐量分析表明：假定一个无线多跳网络由 n 个节点组成，节点密度恒定，源节点和目的节点分散在网络中，可空间复用，那么该网络的总吞吐量（容量）等于 $O(\sqrt{n})$ 。这就意味着网络总吞吐量随着网络覆盖区域的增大而提高。但是在 WSN 中，由于所有源节点只有一个共同的目的节点（中心节点），所以 WSN 的容量不会随着网络覆盖范围的增大而提高。WSN 的容量是中心节点每秒能够接收的最大比特数，等于中心节点一跳范围内的节点能够成功发送的比特数。这些一跳相邻节点给一个公共目的节点发送，因而不能实现空间复用。因此，WSN 的总吞吐量等于 $O(1)$ ，这就意味着 WSN 总吞吐量不会随着网络覆盖范围的增大而提高。

由此得到一个结论：WSN 总吞吐量受到中心节点一跳范围内所使用的 MAC 协议的容量的限制。因为漏斗-MAC 协议是一种 CSMA 和 TDMA 的混合协议，所以需要分别考虑这两种 MAC 协议的容量。

参考文献[22]对 CSMA 的性能分析结果表明：信道饱和（每个节点总是有信息需要发送）时 CSMA 的容量为

$$S_c = f(n, W_{\min} \times m) \quad (4-8)$$

式中， n 表示竞争节点数量， W_{\min} 表示最小竞争窗口大小， m 表示最大退避时间。

信道饱和时 TDMA 的最大信道利用率为

$$S_t = f(t_s, E_p) \quad (4-9)$$

式中， t_s 表示时隙长度， E_p 表示平均分组长度的。

对于漏斗-MAC 协议， W_{\min} 、 m 、 t_s 、 E_p 均是恒定的。一个给定 WSN 的漏斗-MAC 协议的吞吐量为

$$C_f = r \times S_t \times (A_1/A) + (1-r) \times S_c \times [B_1/(B_1+B_2)] \quad (4-10)$$

式中， r 表示 TDMA 帧在超帧中所占的比例， A_1 表示分配给中心节点一跳相邻节点的时隙数量， A 表示所分配的时隙总数量， B_1 表示分配给中心节点一跳相邻节点的发送机会的大小， B_2 表示分配给中心节点两跳相邻节点的发送机会的大小。式（4-10）中第一项 $r \times S_t \times (A_1/A)$ 表示 TDMA 帧的容量， A_1/A 表示中心节点一跳相邻节点参与 TDMA 而获得的容量部分。式（4-10）中第二项 $(1-r) \times S_c \times [B_1/(B_1+B_2)]$ 表示 CSMA 帧的容量， $B_1/(B_1+B_2)$ 表示中心节点一跳相邻节点参与 CSMA 而获得的容量部分。根据漏斗-MAC 协议的时隙分配规则， A 和 A_1 可以计算如下：

$$A = w \cdot \sum_{i=1}^d i \cdot N_i \quad (4-11)$$

$$A_1 = w \cdot \sum_{i=1}^d N_i \quad (4-12)$$

式中, d 表示强度区深度 (转发跳数), N_i 表示中心节点 i 跳相邻节点数量, w 表示所监视流量的加权函数。加权函数 w 对于所有聚集路径都是相同的, 因为在网络饱和时每个节点总是有信息需要发送。在式 (4-10) 中, 只有 A_1/A 是 d 的函数, 其他变量与 d 无关。因此根据式 (4-10), 吞吐量 C_f 取最大值时的 d 等于 A_1/A 取最大值时的 d 。根据式 (4-11) 和式 (4-12) 可得到 $d=1$ 时 A_1/A 最大。因此, $d=1$ 时漏斗-MAC 协议的吞吐量 C_f 最大。从 1 开始进一步增大 d , 吞吐量反而越来越低。这个结果或许令人吃惊, 但是假如考虑漏斗效应, 那么中心节点一跳相邻区域是网络瓶颈, 最需要无竞争的访问协议。

现在考虑网络未饱和情形, 此时吞吐量不再是一个重点关心的问题, 感兴趣的是减少网络碰撞。因而自然想到最大程度使用 TDMA 可以使碰撞达到最小。因此, 在信道未饱和时, 增大强度区深度 d 。但是在漏斗-MAC 协议中, 由于每条聚集路径的转发跳数随着深度 d 的增大而增大, 所以一个超帧需要的时隙数随着深度 d 的增大而增大, 见式 (4-11)。所需时隙数超过可用时隙数, 此时在 TDMA 帧中发生分组丢失。式 (4-11) 给出了一个超帧所需时隙数与深度 d 的关系, 一个超帧中的可用时隙数 t_f 计算如下:

$$A_{\max} = r \times (t_f / t_s) \quad (4-13)$$

因此, 碰撞与传输时间安排溢出之间需要折中平衡, 所以漏斗-MAC 协议在深度 d 时的分组丢失率为

$$L_f = P_s \times (S / (S + S')) + P_c \times (S' / (S + S')) + P_e \quad (4-14)$$

式中, P_s 表示传输时间安排溢出引起分组丢失的概率, P_c 表示碰撞引起分组丢失的概率, P_e 表示其他原因 (比如信道误码、系统出错等) 引起分组丢失的概率, S 表示漏斗区内已安排传输数量, S' 表示漏斗区外竞争传输数量。若 $A < A_{\max}$ 则 $S=A$, 若 $A > A_{\max}$ 则 $S=A_{\max}$ 。

若 $A < A_{\max}$, 则 $P_s = 0$, 所以分组丢失率 L_f 随着深度 d 的增大而下降。在某个值上, 深度 d 的增大导致 $A > A_{\max}$, 此时 $P_s = (A - A_{\max}) / A_{\max}$, 且随着深度 d 的增大而增大。

考虑 $A = A_{\max} + \Delta$ 和 $A = A_{\max}$ 两种情形, 此时均有 $S = A_{\max}$ 。变量 Δ 是正整数, 用于帮助理解对所分配时隙加法递增的响应。两种情形下的分组丢失率之差为

$$\begin{aligned} L_{\Delta} &= (\Delta S) \times ((S + \Delta) / (S + S')) + P_c \times ((S' - \Delta) / (S + S')) - P_c \times (S' / (S + S')) \\ &= (\Delta / (S + S')) \times (1 + (\Delta / S) - P_c) \end{aligned} \quad (4-15)$$

对于任意 $\Delta > 0$ 有 $P_c \leq 1$ (P_c 是概率, 其取值范围在 $[0, 1]$ 内), 所以 $L_{\Delta} > 0$ 。因此, 当 $A > A_{\max}$ 时, 分组丢失率 L_f 随着深度 d 的增大而提高。因此, 当 $A = A_{\max}$ 时, 分组丢失率 L_f 最小, 联合解式 (4-11) 和式 (4-13) 可得到最佳深度为 d_{opt} 。当深度 $d = d_{\text{opt}}$ 时, 漏斗-MAC 协议的分组丢失率小于纯 CSMA 是有保证的, 并且漏斗-MAC 协议和纯 CSMA 的分组丢失率之差等于 $(P_c \times S) / (S + S')$ 。

2. 动态深度调整算法

根据上述分析, 提出如下动态深度调整算法: 假设 A 表示所安排的时隙总数量, A_{\max} 表示一个超帧中可用的最大时隙数量, d_{\max} 表示深度 d 的上限值; 那么中心节点选择网络饱和时 $d=1$, 即使 $d=1$ 也仍然满足 $A > A_{\max}$, 并且假如网络没有饱和, 那么中心节点逐渐增大 d , $A < A_{\max}$, 当 $A > A_{\max}$ 或者 $d > d_{\max}$ 时停止增大 d 。由于深度 d 受中心节点信标信令发射功率的控制

制, 所以存在与中心节点最大可用发射功率匹配的上限深度 d_{\max} 。前面已经验证了如下结论: 当 $A=A_{\max}$ 时, 深度达到最佳值, 网络达到最大吞吐量和最小分组丢失率。这个分析结果证明发射功率的调整方法达到最优。

动态深度调整算法的实际操作如下: 中心节点启动时, 选择与网络中普通传感器节点相同的发射功率, 即中心节点和所有传感器节点采用一个公共发射功率。中心节点监视信道, 计算传输时间安排为 A 。此时可能发生两种不同的情况: $A \leq A_{\max}$ 和 $A > A_{\max}$ 。若 $A > A_{\max}$, 则中心节点不增大下一个信标的发射功率; 若 $A < A_{\max}$, 则中心节点将下一个信标的发射功率增大一个数量级, 监视信道性能。中心节点按照这种方式继续增大信标的发射功率, 直到 $A > A_{\max}$ 或者达到装置限制的最大功率为止。若 $A > A_{\max}$, 则中心节点将下一个信标的发射功率递减一个数量级。若发射功率达到最大值且 $A < A_{\max}$, 则中心节点将发射功率保持在最大值。中心节点连续执行这个动态深度调整算法, 相应调整信标发射功率。动态深度调整算法的伪码见图 4-5 中的“中心节点动态深度调整”。

4.2.7 漏斗-MAC协议的测试床实验评估

本节将采用实验方法评估漏斗-MAC 协议, 介绍各种系统条件下的漏斗-MAC 协议性能实验, 比较漏斗-MAC 协议、B-MAC 协议和 Z-MAC 协议的性能。

1. 实验建立

在 Mica2 传感器节点上实现漏斗-MAC 协议, 采用 TinyOS 默认的 MintRoute 路由协议和 Surge 应用来驱动不同的源速率。Mica2 传感器节点的电台接口速率为 19.2 kb/s。实验测试床是由 45 个传感器节点密集构成的一个栅格, 布置在一个大实验室内, 其配置如图 4-4(d) 所示, 特别说明除外。设置的节点间距和传感器发射功率满足一跳相邻节点实现 80% 以上的分组交付率, 而两跳相邻节点实现 20% 以下的分组交付率。这样就为实验建立起一个相当严格而密集的多跳无线环境。采用 TinyOS 的默认分组长度 (36 B)。

在 B-MAC 协议上面实现漏斗-MAC 协议, B-MAC 协议作为基准 CSMA 系统。关心 B-MAC、Z-MAC 以及漏斗-MAC 在一个真实无线网络环境中的性能表现及其对比, 这个环境的时变无线信道条件影响收敛、链路质量、聚集路径, 所以没有像 Z-MAC 协议那样采用固定路由。对于 B-MAC 协议和 Z-MAC 协议分别采用其默认设置。漏斗-MAC 协议的参数设置如表 4-3 所示, 表 4-3 中未列出的参数设置与 B-MAC 协议相同, 这是因为漏斗-MAC 协议位于 B-MAC 协议上面。对于所有实验, 关闭低功率侦听 (LPL)、B-MAC、Z-MAC 以及漏斗-MAC 采用相同长度的前导, 以便于公平比较。将传感器节点的数据发送功率调整为 -10 dBm , 以便建立一个严格多跳网络 (高达 5 个转发跳)。漏斗-MAC 协议动态调整中心节点信标和传输时间安排的发送功率, 调整范围为 $-10 \sim 5\text{ dBm}$ (即 CC1000 收发信机的最大发射功率), 调整步长为递增或者递减 1 dBm (即一个单位功率等级)。

最初根据传感器节点的时钟精度和安排媒介传输时间所要求的同步精度计算信标发送间隔时间。在信标发送间隔时间可变条件下进行若干次实验, 根据实验确定信标发送间隔时间为 20 s 时得到该精度下的最佳吞吐量。采用惰性信标发送法进行实验, 平衡性能而得到较大的信标发送间隔时间。从实验中观察到: 信标发送间隔时间可以高达 50 s, 只是在 50 s 时吞吐量稍有下降。但是信标发送间隔时间大于 50 s 时, 中心节点吞吐量急剧下降约 30%, 这

说明传输时间安排漂移及其精度损失对于进一步降低信令开销的成本太高。在下面的实验中，信标发送间隔时间为 20 s，以便提高传输时间安排精度，排除任何可能的传输时间安排漂移。表 4-3 表示漏斗-MAC 协议测试床的一组实验参数，在所有实验中都采用表 4-3 所示的参数。

表 4-3 漏斗-MAC 协议实验参数

参 数	取 值
默认数据发射功率 (C_{data})	-10 dBm
信标和传输时间安排的发射功率 ($C_{control}$)	-10~5 dBm
功率动态调整步长 (C_{unit})	1 dBm
信标发送间隔时间 (t_b)	20 s
超帧长度 (t_f)	1 s
时隙长度 (t_s)	30 ms
移动平均因数 (α)	0.9

2. 深度调整的影响

我们对强度区深度对 WSN 测试床吞吐量的影响感兴趣，理由如下：首先，为了验证将 TDMA 区（即强度区）推进到最佳深度之外只会降低中心节点吞吐量；其次，为了说明动态深度调整算法在真实传感器测试床中实现是有效的。为了比较动态深度调整算法与简单的中心节点最后一跳（离中心节点最近一跳的节点）的传输时间安排，将动态深度调整算法固定在一跳上。前面的结果指出：大部分分组丢失发生在离中心节点最近一跳的节点上。根据这个逻辑，考虑一个将强度区深度固定为 1 的“基准算法”，该算法只安排中心节点最后一跳的传输时间安排；考虑一个“优化算法”，使用全使能动态深度调整算法安排其他若干跳的传输时间安排。下面将说明优化算法的性能比简单基准算法好得多。

为了观察深度对性能的影响，将信标发射功率分别设为-10 dBm、-8 dBm、-6 dBm、-4 dBm、0、4 dBm。强度区深度近似为所用信标发射功率的函数。实际上，在栅格实验网络中，可以用信标发射功率覆盖距离（即离中心节点的转发跳距离）来近似表达深度。例如，假如中心节点使用普通传感器节点的默认发射功率发送一个信标，那么深度近似为离中心节点一跳覆盖范围。同样地，希望采用较高发射功率发送信标，具有大于一跳的较大覆盖范围。使用每个信标发射功率设置来观察的性能指标就是吞吐量。将中心节点阻塞点吞吐量定义为 中心节点每秒时间内接收到的比特数。在这些实验中，所有 44 个传感器节点都是源节点。对以下三种源速率分别进行实验：低速（0.2 pps），中速（1 pps），高速（2 pps）。

实验结果如图 4-7（a）所示。对每种源速率，测试中心节点吞吐量随着信标发射功率（近似为强度区覆盖范围的深度）增大的变化情况。实验结果表明：存在一个近似的最佳信标发射功率（即最佳深度），此时吞吐量最大。如果使用的发射功率大于最佳信标发射功率，那么中心节点吞吐量反而下降。这就意味着：假如采用较大发射功率，增大 TDMA 区且大于最佳深度，那么结果是吞吐量下降。

图 4-7（a）证明动态深度调整算法有效。根据前面强度区最佳深度的分析结果，网络饱和条件下的最佳深度接近一跳（即信标发射功率等于传感器节点数据事件发射功率），而网络未饱和条件下的最佳深度大于一跳。实际上，假如将源速率设为 2 pps，这个速率使网络接近饱和，那么实验结果是，最佳信标发射功率为-8 dBm，无线覆盖范围接近一跳（即传感器节

点数据事件发射功率-10 dBm)。从图 4-7 (a) 中可观察到：网络未饱和时的最佳深度大于一跳（即 1 pps 时为 0 dBm，0.2 pps 时大于等于 4 dBm）。这些实验结果验证了前面强度区最佳深度的分析结果，因此为动态深度调整算法建立了可靠基础。

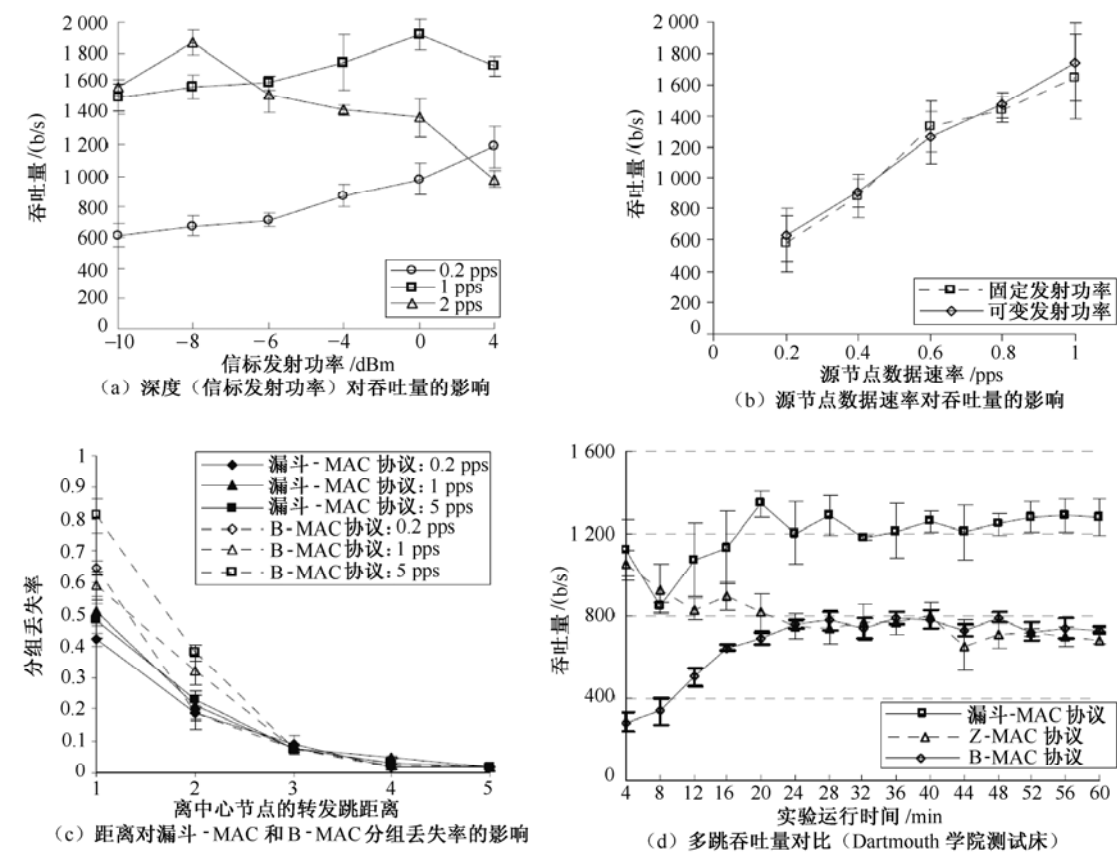


图 4-7 漏斗-MAC 协议的性能

3. 边界节点干扰的影响

下面说明 Meta 传输时间安排广告对于处理前面讨论的干扰问题是有效的。研究强度区深度突然变化对边界节点操作以及中心节点吞吐量的影响。在这个实验中，Meta 传输时间安排广告利用无线媒介的广播特性，节点通过旁听相邻节点发送的、包含嵌入式 Meta 传输时间安排的数据事件分组来实现嵌入式 Meta 传输时间安排的接收。采用 Meta 传输时间安排后允许强度区内使用 TDMA，同时强度区外使用纯 CSMA。

评估几种干扰情形下强度区边界上节点的表现。建立一个实验来研究边界可变性的影响。在这个实验中，中心节点轮流使用两个发射功率-6 dBm、-8 dBm 来改变每个信标的发射功率。选择这两种发射功率是为了使强度区边界近似跨越栅格测试床中心，栅格测试床中心节点密度较高，TDMA 传输时间安排包括栅格测试床中心（在-8 dBm），然后强度区边界离开栅格测试床中心（-6 dBm），处于强度区外面，操作不需要成帧和定时信息，如图 4-4 (d) 所示。

对多种不同源数据速率在-6 dBm、-8 dBm 之间轮流切换实验。图 4-7 (b) 表示各种源数据速率及其相应中心节点吞吐量之间的变化关系，其中 44 个传感器节点全部为源节点。做

两个实验：一个是可变发射功率实验，即信标发射功率在-6 dBm 和-8 dBm 二者之间轮流交换；另一个是固定发射功率实验，将信标发射功率固定为-7 dBm（即-6 dBm 和-8 dBm 的平均值）。由两种实验得到的吞吐量对比如图 4-7（b）所示。对每种源数据速率做 5 次实验，按照 95%的置信区间计算吞吐量。从图 4-7（b）中可以看到：可变发射功率、固定发射功率两种实验得到的吞吐量几乎相同（即相互处在对方的置信区间内）。这个结果说明本实验强调的边界可变性对漏斗-MAC 协议稳定操作能力影响甚小。在实验中记录如下传感器节点：发生信标超时问题的传感器节点，没有成帧信息但是又旁听到 Meta 传输时间安排的传感器节点。在实验中发现 8%的边界节点属于这种节点，也就是说，随着信标发射功率在-6 dBm 和-8 dBm 二者之间轮流切换，强度区内外的节点始终是一致的。这就说明这 8%的传感器节点如果没有成功旁听到嵌入式 Meta 传输时间安排广告，则可能成为干扰节点。

4. 分组丢失率分布

前面对漏斗效应对 B-MAC 协议分组丢失率的影响进行了定量讨论。下面评估漏斗效应对漏斗-MAC 协议的影响。采用相同的实验环境（即 45 个传感器节点的多跳测试床）和性能指标（分组丢失率），如图 4-4（c）所示。实验结果如图 4-7（c）所示。为了比较，图 4-7（c）还包括 B-MAC 协议的分组丢失率结果。图 4-7（c）表示漏斗-MAC 和 B-MAC 的分组丢失率随着离中心节点距离的增大的变化关系。比较漏斗-MAC 吞吐量曲线和 B-MAC 吞吐量曲线的倾斜度，可以看到漏斗效应得到减轻。漏斗-MAC 协议和 B-MAC 协议的中心节点前两跳的分组丢失率差距很大，这是因为离中心节点距离小于三个转发跳时漏斗效应是活动的，两条曲线在离中心节点三个转发跳处汇聚在同一点，此时不再存在漏斗效应。大于三个转发跳后，在离中心节点三、四、五个转发跳处，漏斗-MAC 协议和 B-MAC 协议均表现出类似于 CSMA 的性能。但是，漏斗-MAC 协议中心节点前两跳的分组丢失率比 B-MAC 协议小得多。例如，在源节点数据速率为 4 pps 时，B-MAC 协议的一跳分组丢失率为 81%、两跳分组丢失率为 40%，而漏斗-MAC 协议的一跳分组丢失率降低到 48%、两跳分组丢失率降低到 22%。改变源节点数据速率（0.2 pps、1 pps、4 pps），漏斗-MAC 协议的分组丢失率几乎保持不变，而 B-MAC 协议的分组丢失率变化相当大。

5. 多跳吞吐量

下面在 Dartmouth 学院的 WSN 测试床上进行多跳实验，比较 B-MAC、Z-MAC 和漏斗-MAC 三种 MAC 协议的多跳吞吐量。

图 4-7（d）表示在实验期间跟踪到的 B-MAC、Z-MAC 和漏斗-MAC 三种 MAC 协议的吞吐量，其中所有 44 个节点全部为源节点，其数据速率为 5 pps，这是一个重流量载荷实验方案。做 5 次实验，按照 95%置信区间计算吞吐量。在开始实验时，Z-MAC 和漏斗-MAC 吞吐量表现相同，但是 B-MAC 吞吐量表现最差。值得注意的是，直到实验进行约 20 min 才完全建立从所有源节点到达中心节点的路由。默认设置下，网络性能约在实验进行到 20 min 时才稳定，此前 MintRoute 之类的路由协议需要花费相当长的时间来建立足够的路由状态。一个节点若是没有到达中心节点的路由而将其事件数据发送到广播信道上，则会引起网络拥塞，进一步降低吞吐量。随着越来越多的源节点获得路由和聚集路径，漏斗-MAC 协议、B-MAC 协议的吞吐量随着得到提高。随着实验时间的推进，漏斗-MAC 协议的吞吐

量总是优于 B-MAC 协议。

从图 4-7 (d) 中可以观察到: 随着时间的推进, Z-MAC 协议的吞吐量平稳下降。Z-MAC 协议只是在开始时运行 DRAND 协议, 不是周期性地运行 DRAND。Z-MAC 协议吞吐量下降有可能是因为 Z-MAC 协议易受传输时间安排漂移的影响, 传输时间安排漂移后, 由于时变无线信道状况和可能的路由变化, 由 DRAND 计算出来的起始传输时间安排不再有效, 从而强迫 Z-MAC 协议性能降到 CSMA 性能, 如图 4-7 (d) 中 Z-MAC 协议曲线所示。

参 考 文 献

- [1] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic. Impact of radio irregularity on wireless sensor networks. In *MobiSys '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 125-138, New York, NY, USA, 2004. ACM Press.
- [2] Audio-Video Transport Working Group, H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RFC 1889: RTP: A transport protocol for real-time applications. Jan. 1996.
- [3] H. Balakrishnan. Opportunities in high-rate wireless sensor networking. NSF NOSS Principal Investigator and Informational Meetings, October 2004.
- [4] A. L. Edwards. The correlation coefficient. In *An Introduction to Linear Regression and Correlation*, pages 33-46. W. H. Freeman, 1976.
- [5] A. El-Hoiydi. Spatial TDMA and CSMA with Preamble Sampling for Low Power Ad Hoc Wireless Sensor Networks. In *ISCC*, pages 685-692, July 2002.
- [6] A. Ephremides and O. A. Mowa. Analysis of a hybrid access scheme for buffered users-probabilistic time division. In *IEEE Transactions on Software Engineering*, Vol. SE-8, No.1, pages 52-61. IEEE, Jan. 1982.
- [7] S. Ganeriwal, R. Kumar, and M. Srivastava. Timing-sync protocol for sensor networks. In *Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)*, Los Angeles, CA, November 2003.
- [8] Injong Rhee, Ajit Warrier, Mahesh Aia and Jeongki Min. Z-MAC: a Hybrid MAC for Wireless Sensor Networks. In *Proc. of 3rd ACM Conference on Embedded Networked Sensor Systems (SenSys 2005)*, pp.90-101, November 2005.
- [9] Z-MAC TinyOS Source Code: <http://www.csc.ncsu.edu/faculty/rhee/export/zmac/software/zmac/zmac.htm>.
- [10] A. Warrier and I. Rhee. Stochastic analysis of wireless sensor network MAC protocols. Technical report, Computer Science Department, North Carolina State University, Raleigh, NC, 2005.
- [11] D. L. Mills. Internet time synchronization: The Network Time Protocol. In Z. Yang and T.A. Marsland, editors, *Global States and Time in Distributed Systems*. IEEE Computer Society Press, 1994.

- [12] Jeremy Elson, Lewis Girod and Deborah Estrin. Fine-Grained Network Time Synchronization using Reference Broadcasts. In the proceedings of the fifth symposium on Operating System Design and Implementation (OSDI 2002), December 2002.
- [13] N. Malpani, J. L. Welch, N. Vaidya. Leader election algorithm for mobile ad-hoc networks. In Proceedings of 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communication, pp.96-103, August 2000.
- [14] X. Chen. Dual near field effect in radio frequency simulations. 2002 Summer Computer Simulation Conference, San Diego, USA, July 2002.
- [15] Chipcon Corporation. CC1000 low power FSK transceiver. <http://www.chipcon.com/files/CC1000DataSheet.pdf>.
- [16] D. Petrovic, R. C. Shah, K. Ramchandran, J. Rabaey. Data funnelling: routing with aggregation and compression for wireless sensor networks. In Proc. of IEEE Sensor Network Protocols and Applications (SNPA 2003), May 2003.
- [17] Gahng-Seop Ahn, Emiliano Miluzzo, Andrew T. Campbell, Se Gi Hong and Francesca Cuomo. Funneling-MAC: A Localized, Sink-Oriented MAC For Boosting Fidelity in Sensor Networks. In Proc. of 4th ACM Conference on Embedded Networked Sensor Systems (SenSys 2006), pp.293-306, November 2006.
- [18] Funneling-MAC project webpage: <http://www.cs.dartmouth.edu/~sensorlab/funneling-mac/>
- [19] Funneling-MAC Technical Report: <http://www.cs.dartmouth.edu/~sensorlab/funneling-mac/TAPTR-2006-08-003.pdf>.
- [20] P. Gupta and P.R. Kumar. The Capacity of Wireless Networks. IEEE Transactions on Information Theory, vol.46, no.2, pp.388-404, March 2000.
- [21] J. Li, C. Blake, D. D. Couto, H. Lee, and R. Morris. Capacity of ad hoc wireless networks. ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM 2001), Rome, Italy, July 16-21, 2001.
- [22] G. Bianchi. Performance Analysis of the IEEE 802.11 Distributed Coordination Function. Journal on Selected Areas in Communications, vol.18, no.3, pp.535-547, March 2000.

第 5 章 无线传感器网络数据中心路由协议

5.1 协商式传感器信息分发协议 (SPIN)

协商式传感器信息协议 (Sensor Protocols for Information via Negotiation, SPIN) 是一组基于协商的、适用于 WSN 的信息分发协议。SPIN 重点在于将单个传感器的观测结果数据高效分发到网络中的所有传感器节点, 将所有传感器节点当做可能的中心节点来处理。这样解决问题有几个好处: 首先, 提供一种方法, 用于将对环境的了解完整地复制 (传播) 到整个网络中, 从而提高系统的容错能力; 其次, 提供一种方法, 用于将信息关键内容 (比如已经检测到对监视网络的入侵) 分发给网络中所有节点。

经典泛洪从源节点开始分发, 源节点将其数据发送给自己的所有相邻节点。每个节点接收到数据后, 将其写入自己的存储器, 然后将其复制发送给自己的所有相邻节点。因此, 经典泛洪是直接转发协议, 不需要任何节点的状态, 数据分发迅速, 适用于带宽资源丰富、链路不易于丢失 (中断) 的网络。

经典泛洪存在如下三个缺点, 因而不适用于 WSN:

① 信息传输暴: 在经典泛洪中, 节点总是将数据发送给自己的相邻节点, 而不管相邻节点是否已经接收到其他源节点发送的相同数据。这会引起信息传输暴问题, 信息传输暴问题随着节点密度的增大而线性变严重。

② 重叠: 传感器节点常常覆盖相互重叠的地理区域, 常常收集重复的传感器数据片段。经典泛洪算法将一个数据片的两个复制发送给同一个节点, 既浪费能量, 又浪费带宽。重叠问题比信息传输暴问题更加难以解决, 因为信息传输暴问题只是网络拓扑的函数, 而重叠问题既是网络拓扑的函数, 还是将所观测数据映射到传感器节点的函数。

③ 资源盲目性: 在经典泛洪算法中, 节点不会根据其在给定时间可用能量调整其操作。嵌入式 WSN 必须具有资源意识, 必须使其通信和计算自适应其能量资源状态。

SPIN 协议族综合了两个能够解决这些问题的关键新技术: 协商和资源自适应。

为了克服信息传输暴问题和重叠问题, SPIN 节点首先相互协商, 然后再发送数据。协商有助于确保只有有用信息被发送, 避免将多余的数据发送到整个网络中。但是, 为了协商成功, 节点必须能够描述或者命名其观测到的数据。将 SPIN 协商采用的高级数据描述符称为 Meta-Data。

在 SPIN 协议中, 节点在发送数据前查询其资源。每个传感器节点均有其自己的资源管理器, 用于跟踪资源消耗情况, 应用在发送或者处理数据前查阅资源管理器, 从而允许传感器在能量较低之时减轻执行任务, 比如更加谨慎地转发三方数据。

SPIN 的这些特性克服了经典泛洪的上述三个缺点。发送数据之前的协商过程排除了冗余数据消息的发送, 因而排除了信息传输暴问题。采用 Meta-Data 描述符允许节点命名所观测到数据的所需部分, 从而排除了可能发生重叠问题。SPIN 具有本地能量意识, 从而允许传

感器节点在其能量较低之时减轻所执行的任务，进而延长工作寿命。

5.1.1 SPIN概述

SPIN 协议族依赖两个基本思想：第一，为了实现高效操作和节能，传感器应用需要相互交换其已经获得的以及仍然需要获得的数据；交换传感器数据可能是代价昂贵的网络操作，但是交换有关传感器数据的数据却代价不昂贵。第二，WSN 中的各个节点必须监视、自适应其自身能量资源的变化，以便延长系统的工作寿命。

SPIN 协议设计部分受到应用层帧格式（Application Level Framing, ALF）原理的影响。采用 ALF 时，网络协议必须选择对应用有意义的传输单元，即最好按照应用数据单元（Application Data Unit, ADU）进行分组化处理。ALF 协议的一个重要组成部分是传输协议和应用之间的公共数据命名，SPIN 采用的 Meta-Data 遵循该命名。SPIN 对 ALF 思想作了改进：最好按照应用控制和特定应用方式，不是采用网络拓扑信息而是采用应用数据规划和每个传感器节点的资源状态信息，做出路由决策。这种将命名和路由综合在一起的方法对许多网络都有吸引力，特别是对移动 WSN 非常有吸引力。需要详细了解 ALF 的读者可参阅参考文献[1]。

下面将介绍 SPIN 协议族的各个组成部分以及两个 SPIN 协议，即 SPIN-1 和 SPIN-2。

5.1.2 Meta-Data

传感器采用 Meta-Data 简单而又完整地描述所收集的数据。假如传感器数据 X 的 Meta-Data 描述符为 x ，那么按照字节计算， x 的长度必须小于 X 的长度，这样 SPIN 协议才能发挥效能。假如实际上两个数据是不同的数据，那么其相应的 Meta-Data 描述符也应该不同。同理，两个相同数据的 Meta-Data 描述符应该共享同一个 Meta-Data 描述符。

SPIN 协议没有指定专门的 Meta-Data 格式，Meta-Data 格式是针对具体应用的。各个传感器若是覆盖不相互重叠的地理区域，那么就可以简单使用其自己的 ID 作为 Meta-Data 描述符，因此 Meta-Data 描述符 x 就可以表示“传感器 x 收集的全部数据”含义。照相机传感器可以使用 (x, y, ϕ) 作为 Meta-Data，其中 (x, y) 表示地理坐标， ϕ 表示方向。由于每种应用的 Meta-Data 格式可能不同，所以 SPIN 根据每种应用来解释和综合其自己的 Meta-Data。采用 Meta-Data 后，存在有关 Meta-Data 的存储、恢复、通用管理等方面的开销，但是却可以简单地表示大量数据消息，由此带来的好处远胜于前面提到的那些开销。

5.1.3 SPIN消息

SPIN 节点使用如下三类消息进行通信：

- ① ADV：表示一种新的数据广播消息。一个 SPIN 节点有数据需要发送时，可以播发一条包含 Meta-Data 的 ADV 消息来通知其他传感器节点自己将要发送数据。
- ② REQ：表示申请数据的消息。一个 SPIN 节点想要某种实际数据时发送一条 REQ 消息，用于申请该数据。
- ③ DATA：表示数据消息。DATA 消息包含实际传感器数据以及 Meta-Data 头。

因为 ADV 和 REQ 消息只包含 Meta-Data, 所以 ADV 和 REQ 消息较短, 其发送成本和接收成本比其相应 DATA 消息低。

5.1.4 SPIN资源管理

SPIN 应用具有资源意识和资源自适应能力; 查询系统资源, 以便知道自己能够使用的能量数量; 计算在 WSN 中进行数据计算、数据发送、数据接收的能量开销。SPIN 节点有了这些信息就能够对其资源的高效使用做出正确的决策。SPIN 节点不会为其协议指定专门的能量管理策略, 而是指定一个接口, 应用通过该接口能够获悉自己可用的能量数量。

5.1.5 SPIN实现

SPIN 是一个应用层方法, 用于实现网络通信。因此, 通过定义 API, 按照中间件应用库方式来实现 SPIN 协议。这些库实现 SPIN 协议的基本消息类型、消息处理程序、资源管理函数。传感器应用使用这些库来建立自己的 SPIN 协议。

5.1.6 SPIN-1: 3 步握手协议

SPIN-1 协议是一个简单的握手协议, 用于将数据分发到无损耗网络中。SPIN-1 协议按照三个步骤 (ADV-REQ-DATA) 工作, 每个步骤对应上述三类消息之一。当一个传感器节点获得新的数据并且需要将该数据分发到网络中的时候, SPIN-1 协议就开始工作。SPIN-1 协议给该节点的相邻节点发送一条 ADV 消息, 给新数据命名, 将这一步称为 ADV 阶段。相邻节点接收到 ADV 消息后, 检查自己是否已经接收到该消息或者是否已经申请过该广播数据, 若没有, 则给发送节点回送 REQ 消息来响应 ADV 消息, 表示申请丢失的数据, 将这一步称为 REQ 阶段。当 SPIN-1 协议的启动节点使用 DATA 消息响应 REQ 消息后, SPIN-1 协议操作结束, DATA 消息包含丢失的数据, 将这一步称为 DATA 阶段。

图 5-1 给出一个 SPIN-1 协议例子。节点 B 接收到节点 A 发送来的 ADV 消息后, 检查自己是否已经有了所广播的全部数据。假如没有, 那么节点 B 给节点 A 回送一条 REQ 消息, REQ 消息列出所需要的全部数据。节点 A 接收到节点 B 的 REQ 消息后, 恢复节点 B 申请的数据, 并将其按照 DATA 消息发送给节点 B。节点 B 又给其全部相邻节点发送 ADV 消息, 广播其从节点 A 接收到的新数据。节点 B 知道节点 A 已经拥有该数据, 所以不会给节点 A 发送。然后这些相邻节点给其全部相邻节点播发该新数据, SPIN-1 协议依此连续不断地工作。

在图 5-1 中, 需要注意几个问题。第一个问题是, 假如节点 B 有自己的数据, 那么节点 B 可以将自己的数据和节点 A 的数据累积在一起, 然后将累积数据发送给自己的全部相邻节点。第二个问题是, 不要求节点对协议的每条消息都做出响应。在图 5-1 中, 有一个相邻节点不会给节点 B 发送 REQ 消息。假如一个节点已经拥有所广播的数据, 那么该节点就不会响应消息。

尽管 SPIN-1 协议是为无损耗网络设计的, 但是, SPIN-1 协议可以较容易适应有损耗网络和移动网络。节点采用周期性重播可以补偿丢失的 ADV 消息, 重新申请在固定时间内没有接收到的有关数据内容则可以补偿丢失的 REQ 消息和 DATA 消息。对于移动网络, 本地

拓扑变化会触发更新节点的相邻节点列表。假如一个节点获悉其相邻节点列表已经发生改变，那么自然可以重播其全部数据。

SPIN-1 协议的优点是简单。每个网络节点接收到新数据时很少进行决策，因此计算能耗非常少。此外，每个节点只需知道其一跳范围内的网络相邻节点。运行 SPIN-1 协议不需要其他拓扑信息会带来很重要的效果。首先，SPIN-1 可以在完全未配置的 WSN 中运行，确定启动这种网络最近相邻节点的开销低；其次，假如网络拓扑频繁变化，那么只需将这些拓扑变化信息传播到一跳范围内；然后各个传感器节点可以继续运行 SPIN-1 协议。

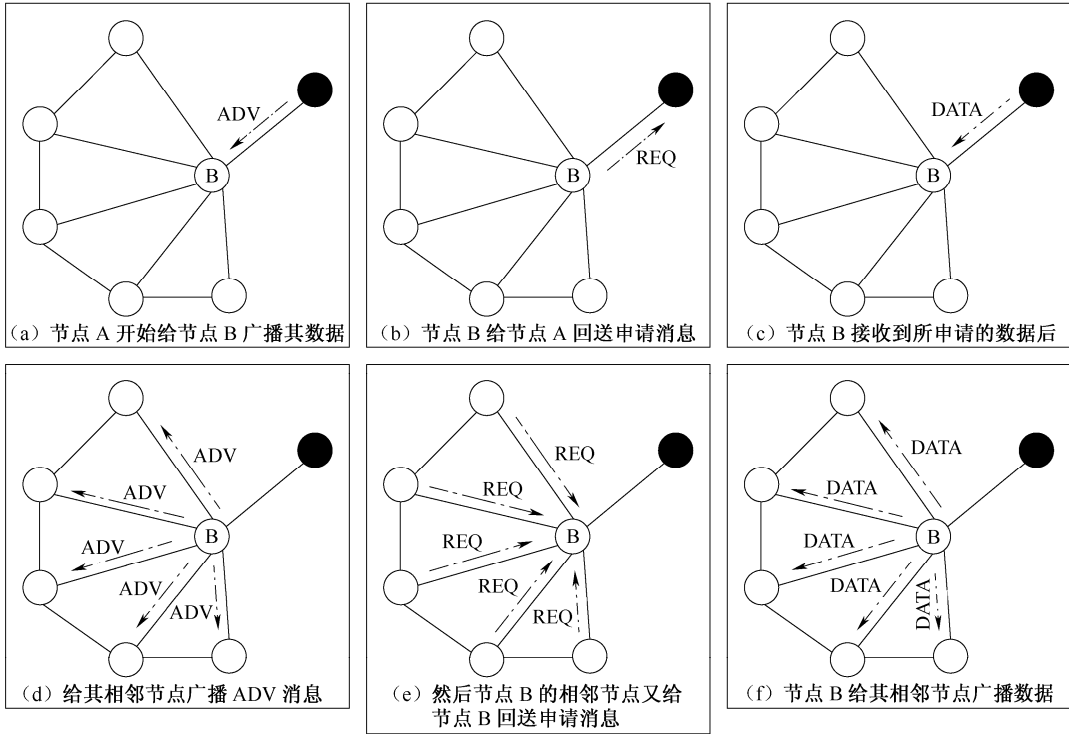


图 5-1 SPIN-1 协议

5.1.7 SPIN-2: 低能量门限的SPIN-1

SPIN-2 就是给 SPIN-1 增加一个简单节能机制后的协议。当能量充足时，SPIN-2 节点像 SPIN-1 节点那样，采用三步骤协议进行通信。一个 SPIN-2 节点观测到其能量快要接近低能门限值时，则减轻其参与 SPIN-2 协议的操作，自适应其能量的变化。一般情况下，一个节点若是相信其能量在进入低能门限之前能够完成协议的其他全部操作，则只参与协议某个步骤的操作。这种节能方式隐含着：假如一个传感器节点接收到新数据并且相信自己有足够能量和其全部相邻节点共同参与协议全部操作，那么该节点只启动三步骤操作协议。类似地，一个节点接收到一个广播，但是却没有足够能量来发送申请和接收相应数据，则不发送申请。这种节能方法不妨碍节点接收消息，因此在能量低于低能门限值时将能量消耗在 ADV 消息或者 REQ 消息上。但是，这种节能方法不允许节点在其能量低于低能门限值时处理 DATA 消息。

5.1.8 用于与SPIN比较的其他数据分发算法

1. 经典泛洪

在经典泛洪算法中，需要将数据分发到网络中的节点开始将该数据的复制发送给自己的全部相邻节点。任何一个节点接收到新数据后，首先做出该数据的复制，然后将该数据发送给自己的全部相邻节点，但是不包括刚刚将该数据发送来的那个相邻节点。一组节点用于接收数据，然后将该数据转发给自己的全部相邻节点所花费的时间称为一个循环时间。当网络中全部节点接收到该数据复制之时，经典泛洪算法运行完毕即收敛。经典泛洪算法需要 $O(d)$ 个循环才会收敛， d 表示网络直径，这是因为经典泛洪算法最多需要经过 d 个循环才会将数据从网络的一端传递到网络的另一端。

尽管经典泛洪算法的简单性可以与 SPIN-1 协议比拟，但是经典泛洪算法既不能解决信息传输暴问题，也不能解决重叠问题。

2. 选择性广播

选择性广播是经典泛洪算法的改进算法，采用随机性来节省能量。选择性广播法不是盲目地将数据转发给所有相邻节点，而是将数据转发给随机选出的一组相邻节点。一个选择性广播节点接收到一个预定相邻节点发送来的数据后，并且随机选出的相邻节点组包含该相邻节点，那么可以将该数据直接回传给这个相邻节点。图 5-2 (a) 说明了选择性广播节点将数据回传给发送节点的理由。假如节点 D 从未给节点 B 转发过数据，那么节点 C 从未接收到该数据。

若采用经典泛洪算法，那么只要数据传递到达高密度节点，就会有较多的数据复制在网络中传递。但是，这些数据复制可能在某个时间点发生传输暴。若采用选择性广播，那么每个节点对每个消息只产生一个复制。复制越少，复制发生传输暴的概率越小。

选择性广播的信息分发速率较慢，但是分发信息消耗能量的速度较慢。考虑一个数据源节点采用选择性广播分发数据。由于源节点只给其一个相邻节点发送数据，而且该相邻节点也只给其一个相邻节点转发数据，所以选择性广播分发数据的最快速率为每轮一个节点。因此，假如网络中有 c 个数据源节点，那么选择性广播分发数据的最快速率为每轮 c 个节点。

选择性广播尽管较大程度地避免了信息传输暴问题，但是却没有解决重叠问题。需要进一步详细了解选择性广播的读者可参阅参考文献[2]。

3. 理想分发

理想分发是一个理想的路由协议，假定全面而完整了解网络以及可能具有最佳性能。图 5-2 (b) 描述一个网络例子：每个节点沿着一条最短路由发送所观测到的数据，每个节点对不同数据的每个部分只接收一次，将这种分发称为理想分发，这是因为观测数据 a 和 c 在可能的最短时间内传递到达每个节点，不会将能量消耗在发送和接收无用数据上。

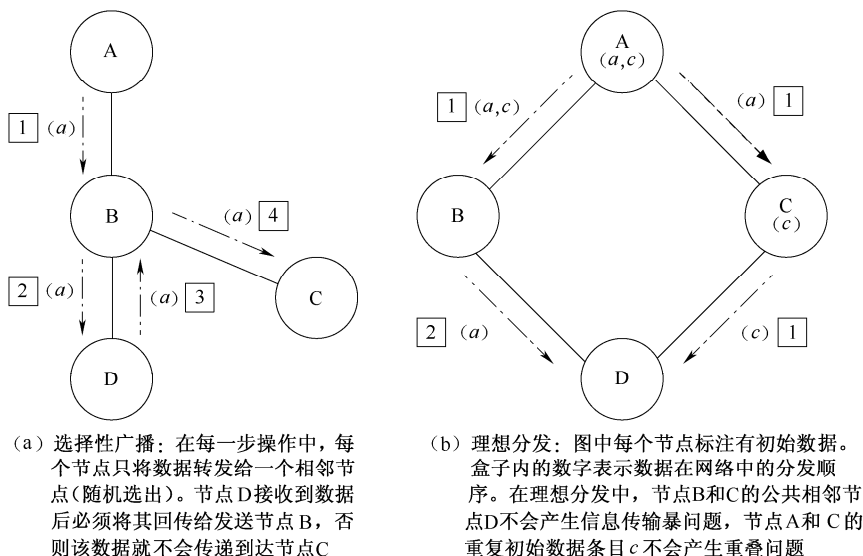


图 5-2 与 SPIN 比较的其他数据分发协议

当前的网络解决方法可以提供几种采用最短路径的信息分发技术。一种方法是网络层多播，如 IP 多目标传输。在这种方法中，网络节点建立和维护分布式特定源节点最短路径树，并承担多目标路由器功能。源节点为了给网络中所有其他节点分发一个新数据，可以将数据发送给一个网络多目标节点组，从而确保数据传递到达参与最短路径路由的全部节点。为了处理传输丢失，修改了理想分发协议，以便使用可靠多目标传输。但是，多目标和特别可靠的多目标都需要依靠复杂的协议机制，其中许多协议操作对于解决 WSN 数据分发中的特殊问题可能没有必要。实际上，在很多方面，SPIN 可以看做应用层多目标协议，将拓扑和数据布置综合到分布式多目标树上。

采用 SPIN-1 修改版仿真理想分发协议的效果。假如跟踪网络中 SPIN-1 协议的消息记录，那么网络中的 DATA 消息就相当于理想分发协议的记录。因此，为了仿真实理想分发协议，运行 SPIN-1 协议，取消 ADV 和 REQ 消息带来的时间开销和能量开销。

5.1.9 SPIN的性能评估

为了评估和比较前面介绍的各种数据分发方法，Wendi Rabiner Heinzelman 博士等人扩展了 ns 软件包的功能，开发了一个 WSN 仿真器。使用这个仿真框架，比较 SPIN-1 和 SPIN-2 与经典泛洪、选择性广播、理想数据分发协议。通过仿真发现：①SPIN-1 协议的吞吐量高于选择性广播和经典泛洪的吞吐量，而 SPIN-1 协议的能耗却比选择性广播和经典泛洪的能耗低得多；②SPIN-2 协议每单位能量交付的数据高于 SPIN-1 协议，自适应 WSN 的能量有限，每单位能量交付的数据接近理想值；③在全部仿真实验中，节点密度较高的节点消耗的能量高于节点密度较低的节点，因此在依靠电池供电的 WSN 中可能存在弱点。

1. ns-2 实现

ns-2 是事件驱动网络仿真器，支持 TCP、路由、多目标等协议的仿真。为了实现 SPIN

类数据分发协议，在 ns 仿真器中增加了几种新的功能。扩充 ns 节点类来建立资源自适应节点，如图 5-3（a）所示。资源自适应节点主要由资源、资源管理器、资源受限应用（Resource-Constrained Application, RCA）、资源受限代理（Resource-Constrained Agent, RCA）以及网络接口组成。资源管理器为应用和各个资源之间提供一个公共接口。RCA 应用负责 SPIN 通信协议和资源自适应决策算法的实现。RCA 代理对 RCA 应用产生的数据封装成分组，然后将其发送给节点的网络接口，以便将其发送给节点的某个相邻节点。

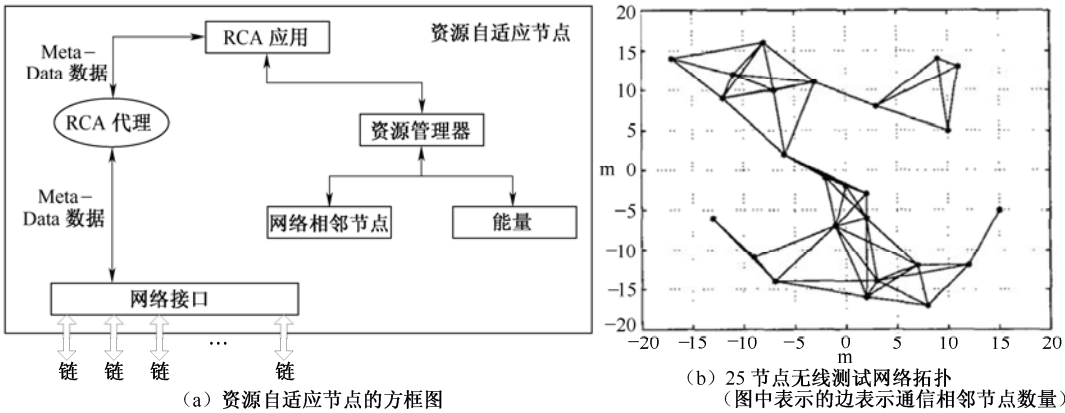


图 5-3 SPIN 的 ns-2 实现框图和仿真网络

2. 仿真测试床

建立一个 25 节点的实验网络，如图 5-3（b）所示。在要求网络图是全连通图条件下，随机产生实验网络，59 条边，节点密度为 4.7 个相邻节点，直径为 8 个转发跳，平均最短路径为 3.2 个转发跳。要求传感器节点无线发射机的发射功率覆盖半径 10 m 以内的全部节点，将这些被覆盖的节点称为该传感器节点的相邻节点。根据当前可用信道数据选择无线传输速率（1 Mb/s）和功耗（发射方式 600 mW，接收方式 200 mW）。消息发送的处理时延从 5~10 ms 之间随机选择。仿真实验中没有考虑 Meta-Data 数据的访问、比较以及管理所产生的时延。从 25 个数据条目中随机选择 3 个数据条目，用于初始化每个传感器节点。这就意味着不同传感器节点的初始数据有重叠，这种情况在 WSN 中经常发生。每个数据条目的长度设为 500 B，每个数据条目有一个互不相同的 16 B Meta-Data 名称。假定实验测试网络不会丢失分组，没有排队时延。表 5-1 概括了实验网络的特性。

在图 5-3（b）所示网络上运行每种协议，按照数据分发速率和能量使用情况跟踪协议运行进程。对于每种实验，对每个协议做 10 次实验，对数据分发次数和能量使用取平均，以便考虑随机处理时延。

表 5-1 25 节点无线测试网络的网络参数配置

节 点 数	25
边	59
平均节点密度	4.7 个相邻节点
直径	8 个转发跳

续表

节 点 数	25
平均最短路径	3.2 个转发跳
天线可达距离	10 m
无线传播时延	3×10^8 m/s
处理时延	5~10 ms
无线传输速率	1 Mb/s
发射功率开销	600 mW
接收功率开销	200 mW
数据长度	500 B
Meta-Data 长度	16 B
网络损耗	无
排队时延	无

3. 能量无限的仿真

第一个仿真实验是假定所有传感器节点有用不完的能量，运行每个数据分发协议，直到协议收敛为止。因为能量是无限的，所以 SPIN-1 和 SPIN-2 是相同的协议。因此，能量无限仿真实验结果只比较 SPIN-1、泛洪、选择性广播、理想数据分发协议。

(1) 时间与数据获取的变化关系

图 5-4 (a) 给出了测试网络在每种协议下随着时间的推移而获得的数据。这些结果曲线明确表明：选择性广播的收敛速度最慢。但是，采用选择性广播，系统在短时间内就已经获得总数据的 85%，大部分时间花费在将剩余 15% 的数据分发给传感器节点。这是因为选择性广播节点将其获得的全部数据发送给一个随机选出的相邻节点。由于选择性广播节点获得的数据很多，而且相邻节点很可能已经接收到所发送数据中的大部分，从而造成很大浪费，因此，选择性广播节点发送数据的开销很大。选择性广播协议保存每个相邻节点的状态，以便使每个传感器节点连续跟踪其已经发送给每个相邻节点的数据，从而减少浪费资源的多余发送，因此选择性广播协议性能得到提高。

图 5-4 (a) 说明 SPIN-1 协议的收敛时间 80 ms，泛洪的收敛时间 10 ms，泛洪的收敛时间比理想协议的长。尽管从收敛时间来比较，SPIN-1 协议比泛洪差许多，但是 SPIN-1 协议的收敛时间与泛洪协议的收敛时间之差是一个常数，与仿真时间长短无关。因此，对较长时间的仿真，SPIN-1 协议比泛洪协议增加的收敛时间可以忽略不计，其原因稍后有详细讨论。

仿真实验结果表明四种协议的数据分发曲线都是上凸曲线。因此推测：通常这些数据分发曲线可能是上凸曲线，而与网络拓扑无关。假如能够推测这些曲线的形状，那么就能够得到协议在不同网络拓扑下的性能表现的某些启示。节点 i 每轮接收到的数据量 d 只依赖离该节点 d 跳远的相邻节点数量 $n_i(d)$ 。但是，因为不同节点 i 、不同距离 d 的 $n_i(d)$ 不相同， $n_i(d)$ 完全依赖具体协议，所以实际上从这些曲线中不能得到通用结论。

(2) 时间与能量消耗的变化关系

在前面的实验中，同时还测量 WSN 随着时间的推移而消耗的能量，如图 5-4 (b) 所示。这些结果曲线表明：选择性广播协议开销最大，完成相同的任务需要比另外两种协议消耗多得多的能量。根据前面所述，在选择性广播协议中增加少量的状态信息，那么选择性广播协

议的系统总能耗将会大幅度下降。

图 5-4 (b) 还说明：泛洪的能耗大约是 SPIN-1 协议能耗的 3.5 倍。因此，SPIN-1 协议通过在收敛时间上的恒定少许牺牲，达到和实现系统能耗的大幅度下降，其原因在于 SPIN-1 协议不会发送无用的大约 500 B 数据条目。

图 5-4 (c) 表示一定节点密度下每个节点的平均能耗。图 5-4 (c) 表明：对于所有协议，每个节点的能耗依赖其节点密度。节点能耗与节点密度的相互影响关系是：假如一个高密度节点碰巧处在一条关键网络路径上，那么该节点可能比其他节点提前失效而停止工作，造成网络分割。处理这种情况是改善四种协议最重要的一个方面。能量无限仿真实验的主要结果概括成表 5-2。

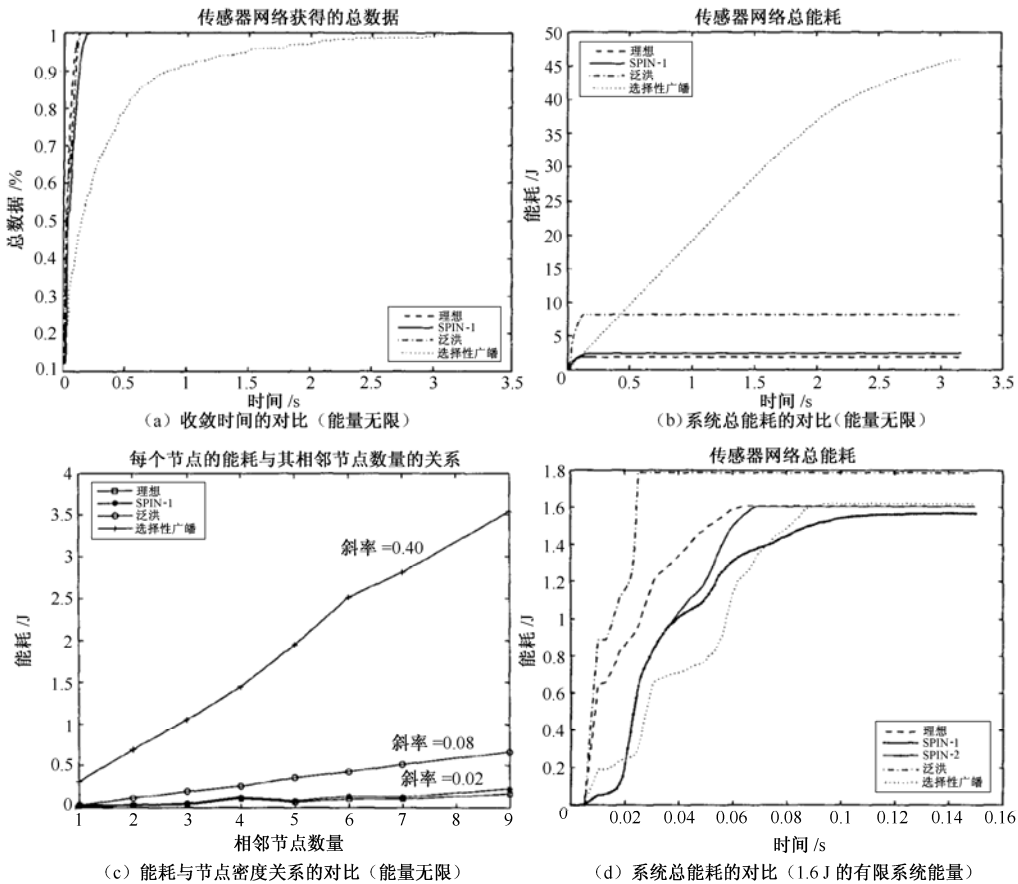


图 5-4 SPIN 的性能

表 5-2 能量不受限制仿真实验的主要结果

相对于理想协议的性能	协 议		
	SPIN-1	泛 洪	选择性广播
增加的能耗/J	0.45	6.3	44.1
增加的收敛时间/ms	90	10	3 025
能耗与节点密度相关曲线的斜率	1.25	5	25
多余数据消息所占比例	0	77%	96%

4. 能量有限的仿真

图 5-4 (d) 给出了在能量有限仿真实验中的能耗速度。图 5-4 (h) 表明：泛洪将能量用尽的速度最快，而选择性广播协议、SPIN-1、SPIN-2 的能耗速度较慢，因此能够保持较长时间的工作。

5. 最佳收敛时间

在很多情况下，需要较多关心协议收敛时间的长短问题，而较少关心协议性能随着时间变化的表现。为了研究这个问题，建立一组实验，用于测试各种网络参数对各种数据分发协议收敛时间的影响。正如前面的实验一样，仿真实验和分析没有考虑排队时延和网络损失，因此是理想网络的最佳实验方案。

图 5-5 表示泛洪、SPIN-1、理想协议的收敛时间随着参数 b （链路带宽）、 d （固定处理时延）、 s （数据长度）在以下实验方案下的变化而变化：①每个传感器节点开始采用单个唯一的数据条目；②每个传感器节点开始采用重叠数据的三个条目。仿真实验条件是 $b=1\text{ Mb/s}$ 、 $d=5\text{ m}$ 、 $s=500\text{ B}$ 。

初始数据中没有重叠条目时，理想协议和泛洪的收敛时间相同。在非重叠条件下，不存在 SPIN-1 协议收敛时间小于泛洪收敛时间的网络参数。但是，对于重叠初始数据，随着链路带宽和每个数据条目长度的变化而存在交叉点。

为了理解这些仿真结果，Wendi Rabiner Heinzelman 博士等人推导出了预测每个协议收敛时间的方程式。对于理想、SPIN-1、泛洪三种协议，任何数据条目传递需要经过的最长路径就是网络最大最短路径或者网络直径 l_d 。在一条带宽为 $b\text{ b/s}$ 链上发送长度等于 s 字节的一条数据消息所需要的时间等于 $8s/b$ 。相对于 DATA 消息的发送时间，ADV 消息和 REQ 消息的发送时间可以忽略不计，因此这里假设不考虑。此外，在发送任何消息（比如 ADV、REQ、DATA）之前，还有固定 $d\text{ ms}$ 和 $[0, r]\text{ ms}$ 之间的一个随机时延之和的网络处理时延。这就意味着理想协议和泛洪协议的收敛时间为

$$l_d(d+(8s/b)) \leq C_{\text{ideal}}, C_{\text{flood}} \leq l_d(d+r+8) \quad (5-1)$$

假如随机时延恒为零，则得到最小收敛时间；假如随机时延总是取可能的最大值，则得到最大收敛时间。典型的收敛时间位于最小收敛时间和最大收敛时间之间的中间值。

可以对 SPIN-1 协议做类似的分析。同样，任何数据条目传递需要经过的最长路径就是网络最大最短路径或者网络直径 l_d 。但是，因为每条消息有 $(d+r)\text{ ms}$ 的处理时延，所以将数据从一个传感器节点传递到下一个传感器节点的时延等于 $3(d+r)+8$ 。这就意味着 SPIN-1 协议的收敛时间范围如下：

$$l_d(3d+(8s/b)) \leq C_{\text{SPIN-1}} \leq l_d(3(d+r)+8) \quad (5-2)$$

因此，当每个节点的初始数据没有重叠并且没有排队时延的时候，SPIN-1 和泛洪（或者理想协议）的收敛时间总是在 $2l_d d$ 和 $2l_d(d+r)$ 之间有一个差值。在泛洪前，SPIN-1 协议收敛时间的网络参数无需选择。但是，SPIN-1 和泛洪协议收敛时间的差值是常数，因此对于长时间仿真可忽略不计。

每个传感器节点初始数据有重叠时，分析方法稍有不同，每个传感器节点从第 $k>1$ 个数据条目开始。开始的时候，数据必须通过的最长路径的长度 l_p 不必等于网络的最大最短路径，但是 l_p 依赖网络布局 and 数据的初始分布。此外，所发送的每条数据消息的长度可以在 s 字节

和 ks 字节之间变化。例如，节点 A 可能将其全部 k 条数据消息发送给自己的相邻节点 B，这些消息长 ks 字节。但是，节点 B 从节点 A 接收到的这 k 条数据消息可能不是全是新数据，因此，节点 B 只将其中的 $k-o$ 条数据消息发送给自己的相邻节点，其中 $0 \leq o \leq k$ 表示节点 A 发送给节点 B 的且节点 B 已经拥有的数据条目数量，因此节点 B 已经给其相邻节点发送了 o 个数据条目。因此，发送一条数据消息的时间位于 $8s/b \sim 8ks/b$ 之间，具体取值依赖数据消息中包含的数据条目数量，所以泛洪和理想协议的收敛时间范围为

$$l_{lp}(d+8) \leq C'_{ideal} \leq l_{lp}(d+r+8k) \tag{5-3}$$

同理，SPIN-1 协议的收敛时间范围为

$$l_{lp}(3d+8) \leq C'_{SPIN-1} \leq l_{lp}[3(d+r)+8k] \tag{5-4}$$

但是，SPIN-1 节点和理想协议节点不会发送无用数据，所以极有可能发送少量的数据条目。因此，SPIN-1 协议和理想协议的收敛时间常常位于上限和下限之间，但是，泛洪协议的收敛时间很有可能接近上限值。假如 SPIN-1 协议收敛时间的下限值比泛洪协议收敛时间上限值小得多，那么 SPIN-1 协议在泛洪协议之前收敛的概率不等于零。当满足下述条件时，SPIN-1 协议在泛洪协议之前收敛

$$l_{lp}(3d+8) << l_{lp}(d+r+8k) \tag{5-5}$$

$$d << 4(k-1) + r/2$$

这就意味着当存在大量初始重叠数据时，由于 SPIN-1 协议发送的数据常常比泛洪协议少且开销低，所以 SPIN-1 协议很可能在泛洪协议之前收敛。

总之，假如每个传感器节点从第一个数据块之后的数据块开始，那么 SPIN-1 协议很可能在泛洪协议之前收敛。但是，假如初始数据是唯一的，那么 SPIN-1 协议永远不会在泛洪协议之前收敛。补充说明：假如每个传感器节点从第 k 个数据块开始且数据是唯一的，那么就等同于传感器节点从唯一数据的第 1 个块开始，重复 k 次单个数据块，因此 SPIN-1 协议永远不会在泛洪协议之前收敛。同理，假如每个传感器节点从非唯一数据的第 1 个块开始，那么就不会出现 SPIN-1 协议或者泛洪协议缩短数据消息的长度，因而 SPIN-1 协议永远不会在泛洪协议之前收敛。

测试床网络的参数如表 5-3 所示。将这些参数代入式（5-3）和式（5-4），得到测试床网络的收敛时间范围为

$$0.063 \leq C'_{ideal}, \quad C'_{flood} \leq 0.154 \tag{5-6}$$

$$0.133 \leq C'_{SPIN-1} \leq 0.294 \tag{5-7}$$

表 5-3 用于计算泛洪、SPIN-1、理想协议的收敛时间范围的网络参数

选 项	符 号	数 值
网络直径/（转发跳数）	l_d	8
重叠初始数据的最短路径长度/（转发跳数）	l_{lp}	7
固定处理时延/s	d	5×10^{-3}
随机处理时延/s	r	5×10^{-3}
初始重叠数据条目数量	k	3
数据长度/B	s	500
链路带宽/(b/s)	B	1e6

实验结果表明：泛洪的平均收敛时间为 135 ms，接近上限值；SPIN-1 协议的平均收敛时间为 215 ms，位于下限值和上限值的中间；理想协议的平均收敛时间为 125 ms。直观上，SPIN-1 协议常常比泛洪协议每条消息少发送 $k=3$ 个数据条目。根据前面的讨论，在给定网络拓扑下，SPIN-1 协议的收敛时间增加量是恒定的，在长时间仿真时可以忽略不计。

若在测试床网络中考虑排队时延，那么泛洪协议的收敛时间比理想协议的长。此外，即使在唯一初始数据条件下，由于过度无用发送导致泛洪节点出现排队时延（而 SPIN-1 节点不会产生这种问题），因而泛洪的收敛时间比 SPIN-1 协议的长。

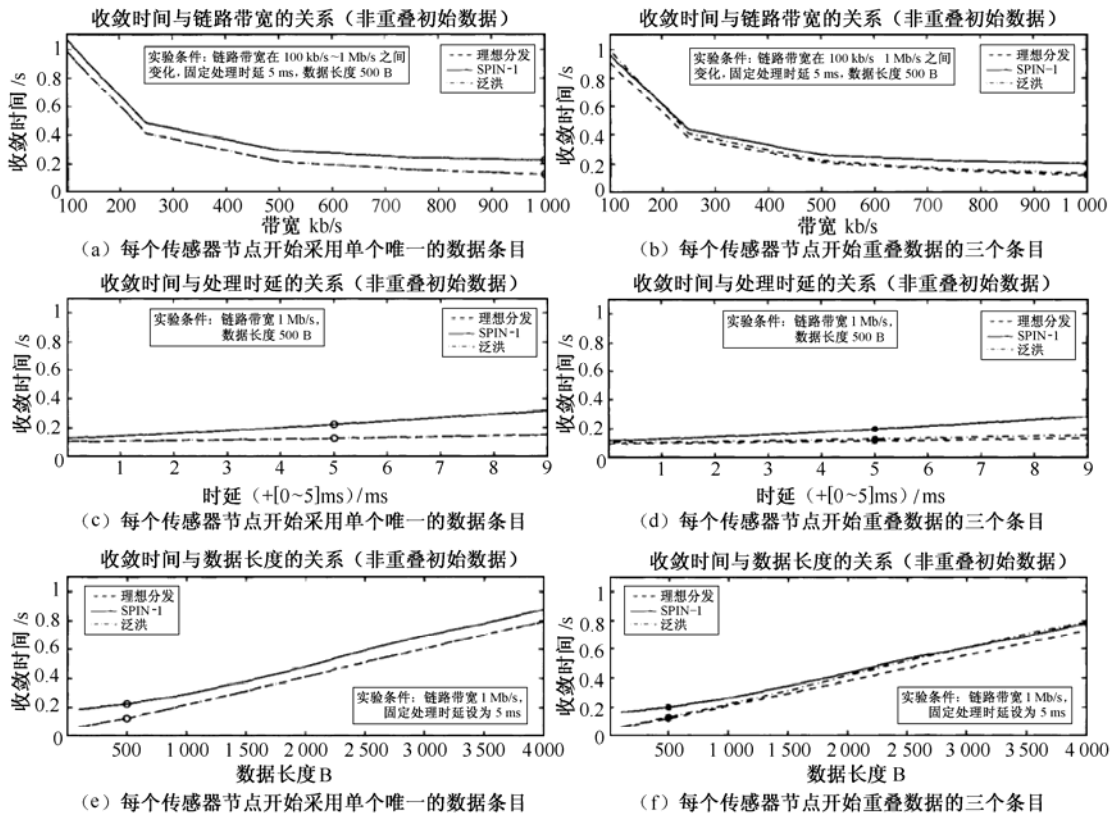


图 5-5 最佳收敛时间

5.1.10 SPIN小结

SPIN 协议族用于 WSN 的数据分发。SPIN 协议采用 Meta-Data 协商和资源自适应来克服传统数据分发方法中的几个缺点。传感器节点采用 Meta-Data 命名，与其他传感器节点相互协商所拥有的数据。相互协商确保只在必需之时发送，确保不会将能量浪费在无用的发送上。传感器节点具有资源意识，只要其资源较少就能够减少其参与的网络操作，从而延长工作寿命。

通过对 SPIN 协议的定量和定性研究，得到以下结论：

- ① 使用 Meta-Data 描述符命名数据和采用 Meta-Data 协商数据发送成功解决了信息传输暴问题和重叠问题。
- ② SPIN-1 和 SPIN-2 是简单的数据高效分发协议，同时不需要维护相邻节点信息。SPIN-1 和 SPIN-2 根据本地邻近信息作出转发策略，因而非常适用于移动传感器环境。

③ 按照时间来衡量，SPIN-1 的性能可以与经典泛洪比拟，在某些情况下甚至优于经典泛洪。按照能量来衡量，SPIN-1 使用的能量只有经典泛洪的 25%；SPIN-2 在协商过程中增加了一个基于门限的资源意识机制，每个单位能量分发的数据比经典泛洪多 60%，并且非常接近每单位能量能够分发的理论数据量。

④ 在全部仿真实验中，SPIN-1 和 SPIN-2 的性能优于选择性广播，在有些条件下的时间性能和能量性能甚至接近理想分发协议。

总之，SPIN 协议从复杂性、能量、计算、通信方面以低成本实现高性能很有前途。

5.2 定向扩散

考虑一个简单的 WSN 工作模型。一个或者多个操作员询问 WSN：“你在地理区域 X 中观察到几个行人？”或者“告诉我区域 Y 中的车辆朝哪个方向行驶”。这些询问会导致给相应区域内的传感器分配任务，从而使这些传感器开始收集信息。各个节点一旦检测到行人或者车辆的行驶方向，就会立即与其相邻节点协作，消除行人位置或者车辆行驶方向上的疑义。然后，这些节点就会将结果报告给操作员。

定向扩散（Directed Diffusion）是数据中心数据分发协议。采用一对属性值命名传感器节点产生的数据。传感器节点通过发送兴趣（Interest）来申请所命名的数据。将与兴趣匹配的数据发送给申请传感器节点。中间节点可以存储或者变换数据，也可以根据所存储的数据引导兴趣的传递。

采用定向扩散可以按照如下方法实现上述传感器例子。将操作员的查询转换成兴趣，然后将兴趣传播给区域 X 或者区域 Y 中的节点。该区域中的传感器节点接收到兴趣后，激活其传感器开始收集有关行人的信息。当传感器节点报告出现行人的信息时，该信息沿着兴趣传递路径的反向路径传递。路径上的中间节点可以累积数据，比如通过组合来自多个传感器节点的报告而更加精确地查明行人的位置。定向扩散的一个重要特性是兴趣和数据的传播和累积由本地交互来决定（相邻节点间或者相邻区域内的消息交换）。

定向扩散不同于 IP 类通信，后者通过端点来识别节点，按照网络内提供的端到端交付服务来分层节点间通信。而定向扩散用于传感器查询分发和处理。使用定向扩散能够实现强壮的多路径交付，经验性自适应一组网络路径，获得大幅度节能，同时中间节点累积查询的响应。

5.2.1 定向扩散的组成要素

定向扩散包含几个要素，采用属性值对数据命名，像数据命名的兴趣那样，将感知任务（或者感知任务的子任务）分发到整个网络中。任务分发建立网络内的梯度（Gradient），网络用于“提取”事件（即与兴趣匹配的数据）。事件开始沿着多条路径朝兴趣源节点方向传递。WSN 强化一条或者少数几条路径。图 5-6 说明了这些要素。

下面针对支持 5.1 节所述任务的特定 WSN，详细描述定向扩散的这些组成要素，该 WSN 执行位置跟踪任务。针对这个定向扩散实例，介绍几种设计方法，详细描述这些设计方法以及该 WSN 的设计。

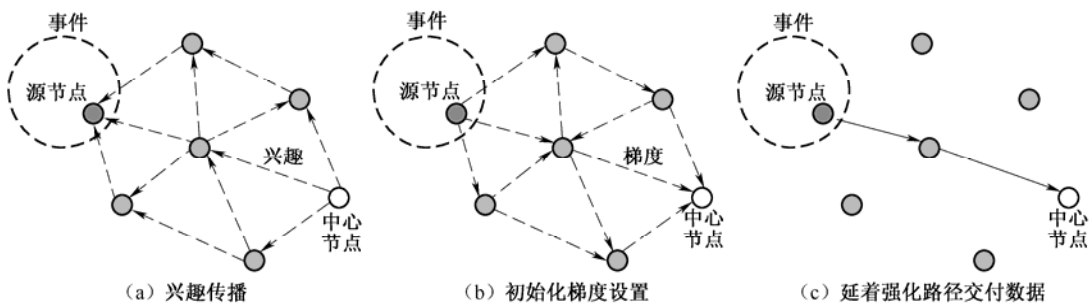


图 5-6 定向扩散的简单示意图

5.2.2 命名

在定向扩散中，使用描述任务的一对属性值命名任务，可以将各种任务的属性值队做成一张表格。前述动物跟踪任务可以简单描述如下（稍后有详细描述）：

```

type = four-legged animal           // 探测四条腿动物的位置
interval = 20 ms                     // 事件回送周期 20 ms
duration = 10 s                      // 下一个 10 s（持续时间 10 s）
rect = [-100, 100, 200, 400]        // 来自矩形场中的传感器

```

为了易于说明，在所定义的坐标系统中采用子域表示一个矩形；在实际中，可能在 GPS 坐标中采用子域表示一个矩形。

直观地，任务描述指定兴趣，以便获取与属性值匹配的数据。正因如此，将一个任务描述称为一个兴趣。采取类似的命名方法命名响应时所发送的数据。例如，一个探测动物的传感器可能产生如下数据：

```

type = four-legged animal           //所看到的四腿动物类型
instance = elephant                  //该类四腿动物实例：大象
location = [125, 220]               //节点的位置
intensity = 0.6                     //信号幅度
confidence = 0.85                   //匹配的可信度
timestamp = 01:20:40                //事件产生时间

```

假定一个 WSN 支持一个任务集，那么设计该 WSN 定向扩散的第一步就是选择一种命名方案。对于所述动物监视 WSN，选择基于属性的简单兴趣和数据命名方案。一般情况下，类型属性的取值范围就是代表移动目标（如车辆、动物、人）的代码集。属性值可以是该范围的任意子集。在动物监视 WSN 中，兴趣中的类型属性取值就是四腿动物的类型代码值。

还有其他的属性值取值范围方法和命名方案。在某种程度上，选择的命名方案可能影响任务的表述，也可能影响扩散算法的性能。

5.2.3 兴趣与梯度

所命名的任务描述包含一个兴趣。通常网络中某个节点（很可能是任意一个节点）给网络注入一个兴趣，该节点就是中心节点。

1. 兴趣传播

假定已经选定命名方案，现在描述如何将兴趣扩散到整个 WSN 中。假定 WSN 中某个节点（即中心节点）提出一个具有特定的 type 和 rect、持续时间 10 min、事件回送周期 10 ms 的任务。参数 interval 说明事件速率；因此，在所述动物监视 WSN 例子中，指定的数据速率为 100 个事件/秒。中心节点记录该任务，并且在持续时间之后清零该任务状态。

对于每个活动任务，中心节点周期性给其每个相邻节点广播一条兴趣消息。初始兴趣包含指定的 rect、duration、interval，但是 interval 取值较大。直观上，初始兴趣可能是试探性的，试图确定是否真正存在检测到四腿动物的传感器节点。为此，初始兴趣指定低速数据速率。在所述动物监视 WSN 例子中，每秒 1 个事件。这不是唯一的选择，但是却表示一种性能平衡考虑。因为不知道源节点的精确位置，所以必须将初始兴趣扩散到比源节点覆盖范围更宽的一个 WSN 区域。结果，假如中心节点选择一个较高的初始数据速率，那么传感器数据分发区域拓宽而可能导致能耗增加。但是，采用较高初始数据速率，缩短了高保真度跟踪的实现时间。初始兴趣采用如下参数设置：

```
type = four-legged animal           // 探测四条腿动物的位置
interval = 1s                       // 事件回送周期 1 s
rect = [-100, 200, 200, 400]       // 来自矩形场中的传感器
timestamp = 01:20:40                // 时:分:秒
expiresAt = 01:30:40                // 本兴趣有效期结束时间
```

兴趣是软状态，中心节点需要周期更新兴趣。为此，中心节点按照时戳单调递增方式简单重发同一个兴趣。由于兴趣在 WSN 中的传递不可靠，所以中心节点周期刷新兴趣是必需的。刷新速率是一个协议设计参数，用于平衡抗兴趣传输丢失而带来的开销。

每个传感器节点维护一个兴趣缓存器。兴趣缓存器中的每个条目就是一个不同的兴趣。在所述动物监视 WSN 例子中，假如两个兴趣的类型不同，或者事件回送周期不同，或者矩形传感器场（可能部分）不相交，那么就认为这两个兴趣不相同。兴趣条目不包含有关中心节点的信息，但是包含直接相邻前一个转发节点的有关信息。因此，根据不同活动兴趣数量来决定兴趣状态。对不同兴趣的定义允许进行兴趣累积。在有些情况下，两个具有相同 type、rect 完全重叠的兴趣 I_1 和 I_2 可以用一个兴趣条目来表示。

表 5-4 定向扩散的基本内容

扩散要素	设计选择
兴趣传播	①泛洪；②基于传感器节点位置的受控泛洪或者定向泛洪；③基于已有存储数据的定向泛洪
数据传播	①强化单路径交付；②多路径交付，可选择不同路径的质量；③多路径交付，概率转发
数据存储和累积	①存在节点失效条件下的强壮数据交付；②感知和数据裁减的协调；③引导兴趣的传播
强化	①强化时间确定规则；②执行强化的相邻节点数量确定规则；③拒绝强化的机制与规则

一个兴趣条目由多个域组成：一个时戳（Timestamp）域、若干个梯度（Gradient）域、一个持续时间（Duration）域。时戳域表示最新所收的匹配兴趣的时戳。一个兴趣条目包含多个梯度域，最多每个相邻节点一个梯度域。每个梯度域包含一个特定相邻节点申请的数据速率域，数据速率可以根据兴趣的事件回送周期推导出来。持续时间域表示该兴趣的近似寿

命，可以根据兴趣的时戳和 `expiresAt` 推导出来。持续时间必须大于网络时延。

一个传感器节点接收到一个兴趣后，检查其兴趣缓存器中是否存有该兴趣。假如该兴趣不存在（根据前面不同兴趣的定义来判断），那么该节点在其兴趣缓存器中建立该兴趣条目。从接收兴趣的说明中得到该条目的参数。该条目有一个特定梯度指向将该兴趣按照特定事件数据速率发送来的那个相邻传感器节点。在所述动物监视 WSN 例子中，中心节点的相邻节点按照 1 个事件/秒的速率建立指向中心节点的梯度建立兴趣条目。为此，必须能够识别出各个不同的相邻传感器节点，可以采用任何本地唯一相邻节点识别码来解决这个问题，比如 IEEE 802.11 MAC 地址、蓝牙分群地址、本地唯一现象识别码。假如缓存器中存在一个没有其发送节点梯度的兴趣条目，那么该节点给该条目增加一个具有特定值的梯度，更新该条目的时戳和持续时间域。假如缓存器中存在一个具有梯度的兴趣条目，那么该节点只需更新该条目的时戳和持续时间域。

当一个梯度达到期满时间时，则应该从其兴趣条目中删除该梯度。注意：并不是全部梯度都是在相同时刻达到期满时间。例如，假如两个不同中心节点表达具有不同期满时间的不同兴趣，网络中某个传感器节点的一个兴趣可能具有多个不同期满时间的梯度。当一个兴趣条目的全部梯度达到期满后，则从兴趣缓存器中删除该兴趣条目。

一个传感器节点接收到一个兴趣后，可能决定将该兴趣重发给其某个相邻传感器节点子集。对于这些相邻传感器节点，该兴趣似乎是该发送节点产生的，尽管该兴趣可能来自某个远处的某个中心节点。这就是一个本地交互的例子。采取本地交互方式将兴趣扩散到整个网络中。并不是所接收到的全部兴趣都会被重传。传感器节点不会重传最近重传过的兴趣。

一般情况下，有几种方法选择相邻传感器节点子集，见表 5-4。选择最简单的方法就是将兴趣重播给全部相邻节点，这等效于将兴趣泛洪到整个网络中。假如不知道哪些传感器节点很可能能够满足该兴趣，那么这种选择是唯一选择。在所述动物监视 WSN 中，可以采用某些位置路由技术运行地理路由协议，这样能够限制兴趣扩散的拓扑范围，从而节省能量。在非移动 WSN 中，节点可以采用已存储的数据引导兴趣的扩散。例如，假如响应一个以前收到的兴趣，即某个节点从相邻节点 A 旁听到位于参数 `rect` 指定的传感器场中某个传感器发送的数据，那么该节点就可以将这个兴趣引导到相邻节点 A，而不是将其广播给全部相邻节点。

2. 梯度建立

图 5-7 (a) 表示将兴趣泛洪到整个传感器场中时建立的梯度。注意：与图 5-6 (b) 不同的是，每对相邻节点建立一个指向对方的梯度。这是本地相互交互的关键结果。一个节点接收到其相邻节点发送来的一个兴趣时，无法知道该兴趣是该相邻节点对自己先前发送兴趣的响应兴趣，还是该相邻节点其他位置上的其他中心节点发送的相同兴趣。相邻节点互设梯度能够保证节点从其每个相邻节点发送的低速事件中接收一个复制。而且，如前所述，这种技术能够使中断路径得到快速恢复，根据经验强化更好的路径，不会发生长时间的闭环。

对于上述动物监视 WSN，一个梯度指定一个数据速率和一个事件的传播方向。一般地，一个梯度指定一个数值和一个方向。对于定向扩散方案，设计者可以自由设置梯度数值。介绍两个梯度的使用例子。图 5-6 (c) 间接描述了二进制梯度。在上述动物监视 WSN 中，梯度有两个决定事件报告速率的数值。在其他 WSN 中，梯度用于沿着不同路径的概率性数据

转发，从而实现某种程度的载荷平衡（见表 5-4）。

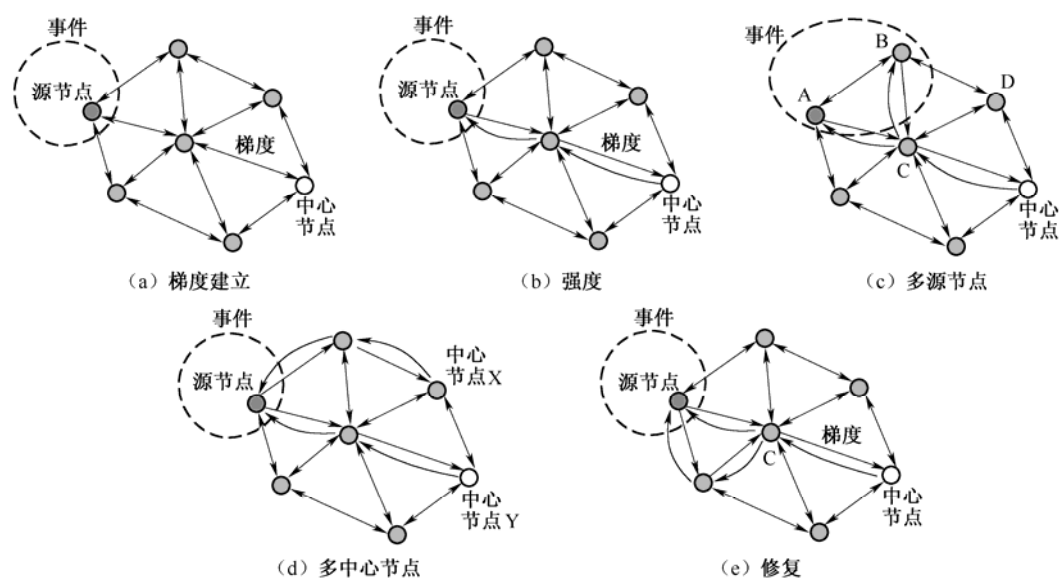


图 5-7 扩散图解

总之，兴趣传播建立网络（或者局部网络）状态，以便易于将数据朝中心节点传送。兴趣传播规则是局部规则，有些类似于某些互联网多目标路由协议中的入网传播。其中一个主要区别是入网传播能够平衡各个单目标路由表、引导入网传播朝源节点进行，但是兴趣传播却没有此功能。

至此已经描述了一种特殊任务类型的兴趣传播规则。一般地，WSN 可以支持多种不同任务类型。兴趣传播规则对于不同任务类型可以不同。例如，“计算矩形区域 R 内下一个 T 秒内所观察到的四腿动物的数量”这类任务不能像所述动物监视 WSN 例子那样支持事件数据速率。但是，兴趣传播规则的一些要素类似于条目存储规则和兴趣重新分发规则。可以从中精选一些类似规则作为每个传感器节点的扩散基础，因此 WSN 设计人员就能够使用兴趣传播技术库（或者就此而言，稍后讨论的数据处理和强化规则）处理不同的任务类型。

5.2.4 数据传播

处在由 rect 确定的区域内的传感器节点按照 5.2.3 节的描述处理兴趣，并且给本地传感器分配任务，以便开始收集样值。简而言之，目标识别算法就是比较采样波形和库中存储的先前采样波形。例如，在所述动物 WSN 例子中，目标识别算法以具有不同于人类的声波或者地震痕迹的四腿动物观测结果为基础。采样波形可能在不同程度上与存储波形匹配，目标识别算法常常将可信度与匹配关联在一起。此外，采样波形的强度可以大致表示信号源的距离，但是不能表示信号源的方向。

一个目标检测传感器节点搜索其兴趣存储器，以便找出相匹配的兴趣条目。因此，一个匹配条目就是其 rect 包围该传感器位置、条目类型与所检测目标类型匹配的条目。若在兴趣存储器中找到一个匹配条目，该节点计算其所有输出梯度之间的最大请求事件速率，然后给

其传感器子系统分配任务，以便其传感器子系统以该最大数据速率产生事件样值。在所述动物监视 WSN 例子中，数据速率初始化为 1 个事件/秒，然后源节点每秒给有梯度的每个相邻节点发送如下形式的一个事件描述：

```
type = four-legged animal           // 所观察到的动物类型
instance = elephant                  // 该类动物的实例：大象
location = [125, 220]                // 节点位置
intensity = 0.6                      // 信号幅度
confidence = 0.85                    // 匹配可信度
timestamp = 01:20:40                // 事件产生的本地时间
```

按照单目标传输方式，将该数据消息有效地发送给相关的每个相邻节点。将该机制应用于无线 MAC 层函数，对性能有重要影响。

一个传感器节点接收到其相邻节点发送来的一条数据消息后，从其兴趣存储器中找出是否有相匹配的兴趣。匹配规则如前所述。假如存储器中没有匹配的兴趣，那么只需简单丢掉该条数据消息。假如存储器中存在匹配的兴趣，那么传感器节点检查和该兴趣条目相关的数据存储器。数据存储器连续跟踪最近观测到的数据条目。数据存储器有几个用途，其中之一就是防止闭环。假如一条所收数据消息存在匹配的数据存储条目，则丢掉该数据消息，否则，即一条所收数据消息不存在匹配的数据存储条目，则将该数据消息存入到数据存储器中，并发给该节点的相邻节点。

传感器节点通过检查其数据存储器就能够确定所收事件的数据速率。为了重发所收数据消息，传感器节点需要检查匹配兴趣条目的梯度列表。假如所有梯度的数据速率都大于或者等于输入事件速率，那么该传感器节点可以将所收数据消息发送给适当的相邻节点。但是，假如有些梯度的数据速率比其他的低（路径强化所致），那么该传感器节点可以下调到适当梯度。例如，考虑一个传感器节点已经在以 100 个事件/秒的速率接收数据，但是其中一个梯度的数据速率是 50 个事件/秒（比如第二个中心节点产生间隔时间较长的、为不同任务而设置的数据速率）。在这种情况下，该传感器节点可以只交替地将每个事件发送给相应相邻节点。另外一种处理办法是，该传感器节点可以采用应用特定方法内推两个连续的事件（在所述动物监视 WSN 例子中，可以选择匹配可信度较高的那个事件）。

预防闭环和速率下调说明具有在所有节点嵌入应用语义的能力，见表 5-4。尽管这种设计与传统网络无关，但是对于特定应用的 WSN 是可行的。这种设计技术将能够大力提高网络性能。

5.2.5 路径建立与路径裁剪的强化

在迄今为止描述的方案中，中心节点开始和反复扩散低事件速率（一个事件/秒）的通知兴趣。这种低速率事件用于路径建立和维护，所以被称为试探性事件。将为试探性事件而建立的梯度称为试探性梯度。源节点一旦检测到匹配目标，就立即向中心节点发送试探性事件（可能沿着多条路径传播）。中心节点开始接收这些试探性事件后，强行增加一个特定的相邻节点，以便得到真实数据（更高数据速率、高质量跟踪目标的事件）。将为接收高质量跟踪时间而建立的梯度称为数据梯度。

1. 采用肯定强化建立路径

通常采用数据驱动本地规则来实现定向扩散的这种新特性，比如一个数据驱动本地规则就是强行增加这样的相邻节点：接收到该相邻节点发送来的并且以前没有接收过的事件。为了强行增加这个相邻节点，中心节点以较短间隔时间（较高数据速率）重发最初的兴趣消息，具体如下：

```
type = four-legged animal           // 探测四条腿动物的位置
interval = 10ms                      // 事件回送周期 10 ms
rect = [-100, 200, 200, 400]        // 来自矩形场中的传感器
timestamp = 01: 22: 35               //时:分:秒
expiresAt = 01:30:40                //本兴趣有效期结束时间
```

相邻节点接收到该兴趣后，知道自己已经有一个指向该相邻节点的梯度，而且还知道发送节点的兴趣指定了一个高于以前的数据速率。假如这个新的数据速率高于现有所有梯度的数据速率（直观上，从这个节点输出的信息增多了），那么该节点必须强行增加至少一个相邻节点。怎么增加？采用数据存储器来实现相邻节点的强行增加。同样采用前面例举的数据驱动本地规则。例如，节点可以选择其第一次收到的、与该兴趣匹配的最新事件的相邻发送节点，也可以选择其最近收到的新事件的全部相邻发送节点。这就意味着只有在发送试探性事件之时才强行增加相邻节点，显然不必强行增加已经正在以较高数据速率发送流量的相邻节点。通过这种按序的本地交互，建立起从源节点到达中心节点的高数据速率事件的传输路径。

然后，数据驱动本地规则根据经验选择一条低时延路径[图 5-7（b）表示中心节点强化路径时得到的路径]。这对于路径质量变化的反应是非常敏感的。只要一条路径交付事件的速度快于其他路径，那么中心节点就会尽量使用这条路径来得到高质量数据。但是，由于接收新事件的触发作用，有可能导致资源浪费。因此，有可能采用较复杂的本地规则（见表 5-4），包括选择接收到大多数事件的相邻节点，或者在其他相邻节点之前持续发送事件的相邻节点。这些选择对路径质量变化的反应具有平衡作用，以便提高稳定性。

2. 建立多个源节点和多个中心节点之间的路径

在前面描述的强化中，似乎间接描述了单个源节点的情形。实际上，所描述的强化规则可以工作在多源节点情形下。为了理解这一点，考虑图 5-7（c）。假设开始所有梯度初始值为低数据速率（即试探性梯度）。根据图 5-7（c）的网络拓扑，从两个源节点 A 和 B 到达中心节点的数据通过两个相邻节点 C 和 D。假如其中一个相邻节点，比如节点 C，一直具有较短时延，那么强化规则只强行增加通过相邻节点 C 的路径[图 5-7（c）中已描述]。但是，假如中心节点通过相邻节点 D 而先接收到节点 B 的事件，但是节点 A 的事件却通过相邻节点 C 而先到达中心节点，那么中心节点尽量从两个相邻节点中获取高质量数据流[图 5-7（c）中未表示]。在这种情况下，中心节点从两个相邻节点中获得两个源节点的数据，其中一个源节点可能是能量低效的。在定向扩散中，中心节点不能关联源节点和事件。因此，“节点 A 的事件”这一表述有点令人误解，其真正含义是节点 A 产生的数据在内容上不同于节点 B 产生的数据。

类似地，假如两个中心节点表达完全相同的兴趣，那么兴趣传播、梯度建立以及强化规则仍然正常工作。不失一般性，假如图 5-7（d）中的中心节点 Y 已经强行增加了一条到达源

节点的高质量路径。但是，其他节点继续接收低数据速率事件（即试探性事件）。当操作人员采用完全相同的兴趣在中心节点 X 给网络分配任务的时候，中心节点 X 可以采用强化规则建立如图 5-7 (d) 中所示的路径。为了确定经验上最好的路径，中心节点 X 不必等待接收到的数据，而是可以利用其数据存储器来立即得到指向自己的高质量数据。

3. 中断路径的本地恢复

迄今为止，已经描述了中心节点触发强化的情况。但是，在定向扩散中，强化路径上的中间节点也可以应用强化规则。这对于中断路径或者降级路径的恢复是有益的。路径中断或者降级的原因包括节点能量耗尽、影响通信的环境因素（如障碍物）。考虑图 5-7 (e)，源节点和节点 C 之间链路的质量降级，其上传输的事件经常被损坏。节点 C 可以采用以下两种方法检测链路降级：一种方法是其上行相邻节点（源节点）给节点 C 发送一个事件，通知到达节点 C 的事件发送速率现在降低了；另一种方法是认识到其他相邻节点已经在发送以前未见过的位置估计。节点 C 检测到链路降级后就可以运用强化规则来寻找如图 5-7 (e) 中所示的路径。最后，节点 C 否定强行增加到达源节点的定向链[图 5-7 (e) 中未表示]。迄今为止，强化规则的描述掩盖了这样一个事实：直接运用强化规则将导致受损链的全部下行节点初始化强化规程。这样最终会选出一条经验上高质量的路径，但是有可能导致资源浪费。这个问题的一种解决方法是节点 C 在其所收事件中内插位置估计，以便下行节点仍然感觉到是高质量跟踪。

4. 采用否定强化裁剪路径

采用肯定强化算法能够得到多条强化路径。例如，在图 5-8 (a) 中，假如中心节点强行增加相邻节点 A，但是又接收到相邻节点 B 发送来的一个新事件，那么中心节点强行增加通过相邻节点 B 的路径，这条路径与通过相邻节点 A 的路径可能相交，也可能不相交。假如通过相邻节点 B 的路径的质量一直较好（即相邻节点 B 先于相邻节点 A 发送事件），那么需要一种机制来否定通过相邻节点 A 的强化路径。

一种否定强化机制是软状态，即对网络中所有数据梯度设置超时时间，但是明确指出强化的数据梯度除外。采用这种方法，中心节点周期性强行增加相邻节点 B 以及停止强行增加相邻节点 A。通过相邻节点 A 的路径上的所有梯度最终下降为试探性梯度。另外一种否定强化的方法（稍后将评估这种方法）是，通过给相邻节点 A 发送否定强化消息，直接降低通过相邻节点 A 的路径的质量。在基于速率的扩散中，否定强化消息就是较低数据速率的兴趣。相邻节点 A 接收到该兴趣后，将其指向中心节点的梯度降级。若其全部梯度都是试探性梯度，那么相邻节点 A 否定强行增加通过已经在以高数据速率给自己发送数据的那些相邻节点的路径。即使通过相邻节点 A 的路径与通过相邻节点 B 的路径部分相交，这个本地规则也仍然起作用。除非两条相交路径被否定强行增加，否则其相交链路不会被否定。这种按序本地交互确保通过相邻节点 A 的路径迅速降级，但是其代价是增加了资源的使用。

为了完善否定强化的描述，需要说明节点使用什么样的本地规则，以便能够决定是否否定相邻节点的强行增加。注意：对于否定强化，这个规则与机制选择是互不相关的。一个可靠选择是否定强行增加那个在 N 个事件或者时间 T 窗口内没有发送新事件的相邻节点（即其他相邻节点一直先于该相邻节点发送事件）。稍后评估的本地规则是基于时间 T 窗口的，在仿真中选择 $T=2$ s。这种规则是比特保守的和能量低效的。例如，即使 10 个事件中有 1 个事

件是首先从相邻节点 A 中接收到的，那么中心节点也不会否定强行增加相邻节点 A。其他规则包括否定强行增加已经很少收到其新事件的那个相邻节点。在决定哪个规则能够实现能量高效全网操作之前需要做大量的实验。

5. 运用否定强化排除闭环路径

否定强化规则除了用于抑制高时延、高损耗路径外，还用于排除路由闭环。闭环路径不能首先交付事件，见图 5-8 (b)。假定只强行增加首先发送试探性事件的相邻节点，那么不会强行增加闭环路径（特别对于单个源节点与单个中心节点之间的路径）。但是，在一轮试探性事件中强行增加的路径可能不同于前几轮试探性事件中强行增加的路径。尽管不会强行增加闭环路径，但是将若个轮强行增加的路径组合在一起，则可能产生闭环路径。一般情况下，采用消息存储技术能够立即抑制闭环消息，但是为了节省资源，裁剪闭环路径还是有用的。但是，这样删除闭环路径并非总是合适的，特别是对于多个源节点和多个中心节点的有些共享高速率梯度图。例如在图 5-8 (c) 中，假如两个源节点发送不同的事件，那么不应该删掉梯度 B—C 和 C—B，这是因为这两个梯度对于交付特定源节点-中心节点对的事件是必需的。这两个梯度尽管可能交付一些闭环消息，但是却总是交付新事件。采用否定强化保守规则，这两个梯度是会被强行增加的。

即使没有闭环路径，维持否定强化规则的保守性也是合理的，这样有用路径就不会被修建。对于图 5-7 (c)，两个源节点可以始终发送不同的事件，但是也可以偶尔发送相同事件。不同源节点产生的相同事件在扩散中被认为是复制事件。事件在扩散过程中与其源节点无关（即节点不能关联源节点和事件）。当前版本的定向扩散只维护高速率路径，总是在高速率路径上发送有用（新）数据，而与源节点无关。因此一般情况下，不能保证从每个源节点至少有一条高速率路径到达每个中心节点（比如当其中有些源节点没有产生有用数据时）。假如否定强化规则过于主动处理复制事件，那么其中一个源节点的路径将被删掉。反之，采用保守规则，则不会否定强行增加某个源节点。

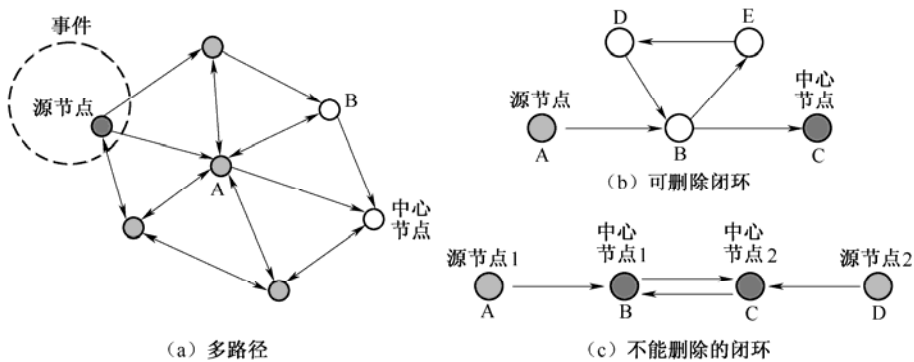


图 5-8 用于路径裁剪和删除闭环的否定强化

5.2.6 定向扩散的分析评估

下面从理论上分析定向扩散、全能多目标 (Omniscient Multicast)、泛洪的数据交付开销。分析模型可以用来检验定向扩散背后直观性的正确性，指出这三种分发协议之间的一些差异。

为了易于分析，在一个极简单理想化设置下分析这三个分发协议。假定一个正方形栅格由 N 个节点组成，节点传输距离保证每个节点在栅格上正好有 8 个相邻节点，如图 5-9 所示。图 5-9 表示能够相互通信的各个节点对之间的通信链。 n 个源节点全部位于栅格左边， m 个中心节点全部位于栅格右边。第一个源节点位于左边中心位置，第 i 个源节点位于第一个源节点上方（ i 为偶数）或者下方（ i 为奇数）的 $d_n \times \lfloor i/2 \rfloor$ 个转发跳处。中心节点也采用这种布置方案，但是相邻中心节点间的距离为 d_m 个转发跳。假定源节点和中心节点只垂直布置，那么必须满足 $\sqrt{N} \geq \max(nd_n, md_m)$ 。

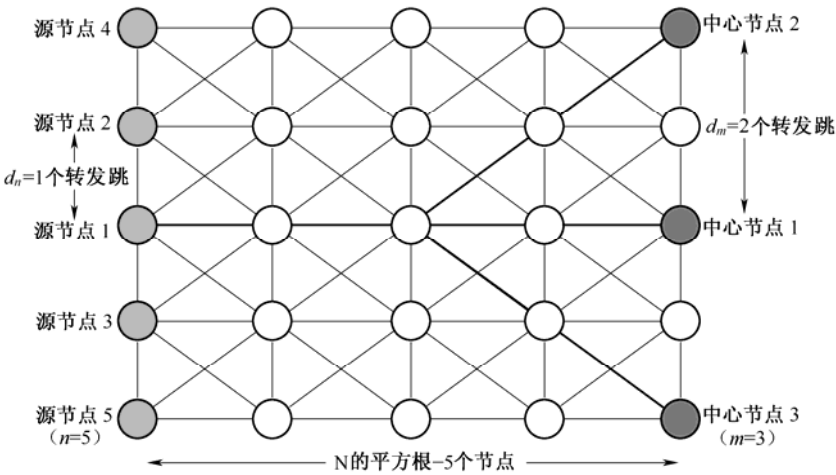


图 5-9 正方形栅格拓扑例子

1. 泛洪

在泛洪方案中，源节点将全部事件泛洪给网络中的每个节点。泛洪作为定向扩散的参考衡量标准。假如定向扩散的能量效率并没有高出泛洪许多，那么认为定向扩散对 WSN 是不适用的。

在分析评估泛洪方案时，考虑的性能是每个源节点发送的一个事件到达所有中心节点的发送和接收的总开销。将开销定义为一个消息发送单位和一个消息接收单位之和。这些假设是对发送和接收的理想假设。发送开销和接收开销可能不相同，并且可能还有其他感兴趣的性能。

根据上述定义，泛洪的开销 $C_f(N, n, m, d_n, d_m)$ （简化为 C_f ）为：

$$C_f = nN + 2n \left[2(\sqrt{N} - 1)\sqrt{N} + 2(\sqrt{N} - 1)^2 \right] = nN + 4n(\sqrt{N} - 1)(2\sqrt{N} - 1) \quad (5-8)$$

因为每个节点对每个事件只发送一个 MAC 广播，所以泛洪 n 个事件（每个源节点泛洪一个事件）的发送开销为 nN 。反之，每个节点能够从所有相邻节点接收到这个事件。因此， n 个事件的接收开销等于网络中链路数（ $\left[2(\sqrt{N} - 1)(2\sqrt{N} - 1) \right]$ ）的 $2n$ 倍。泛洪的数据交付开销为 $O(nN)$ ，渐进高于定向扩散、全能多目标的数据交付开销。

2. 全能多目标

在全能多目标方案中，每个源节点沿着一棵最短路径多目标树将其事件发送给所有中心

节点。理论分析以及随后的仿真实验均没有考虑多目标树建立的开销。全能多目标方案大约表示了在 IP 传感器网络中能够达到的性能。使用全能多目标方案，能够从直观上帮助了解定向扩散机制选择是如何影响性能的。

对于全能多目标方案，数据交付开销是特定源节点最短路径树上链路数量的 2 倍。但是，即使在图 5-9 所示的简单栅格拓扑中，每个源节点-中心节点对之间也存在几条最短路径。采用如下确定性规则选择最短路径：从一个中心节点到一个源节点，只要沿着对角线链能够得到最短路径，那么总是选择对角线链作为下一个转发跳；否则，即沿着对角线链不能得到最短路径，选择垂直链作为下一个转发跳。重复执行这个路径选择规则，直到到达源节点为止。因此，最短路径不包括垂直链。例如，若根部在源节点 j 的一棵最短路径树表示为 T_j ，那么 T_1 上的链路数由两部分组成：垂直链路数 $(\sqrt{N}-1)$ 和对角线链路数 $\left\{d_m \times \frac{m}{2} \times \left(\frac{m}{2} + 1\right) - d_m \times \frac{m}{2} \times [(m-1) \bmod 2]\right\}$ 。其他最短路径选择规则可能得到不同的数据交付开销，因为树上共享链数可能不同。

全能多目标的开销 C_O 等于 n 棵最短路径树开销之和，每棵树的树根在源节点。假如从源节点 j 发送一个事件的开销表示为 $C(T_j)$ ，那么可以得到 $C(T_j) = C(T_1) + C(T_j - T_1) - C(T_1 - T_j)$ 。其中 $C(T_j - T_k)$ 表示沿着所建最短路径树发送和接收的开销再减去从 T_j 开始由 T_j 和 T_k 共享的那些链路上的发送和接收开销。为了易于解释，可以将 $C(T_j)$ 解释为两个开销之和：沿着垂直链发送和接收的开销 $H(T_j)$ ，沿着对角线链发送和接收的开销 $D(T_j)$ 。因此， C_O 可以表示为

$$C_O = \sum_{j=1}^n \left\{ D(T_1) + H(T_j) + D(T_j - T_1) - D(T_1 - T_j) \right\} \quad (5-9)$$

式中

$$H(T_j) = 2 \left\{ \sqrt{N} - 1 - \left(\frac{j}{2} \times d_n - \min \left(\left\lfloor \frac{j}{2} \times \frac{d_n}{d_m} \right\rfloor, \left\lfloor \frac{m - (j \bmod 2)}{2} \right\rfloor \right) \times d_m \right) \right\} \quad (5-10)$$

$$D(T_j - T_1) = 2 \left\{ \left\lfloor \frac{m + (j \bmod 2)}{2} \right\rfloor \times \frac{j}{2} \times d_n + \sum_{l=1}^{\min \left(\left\lfloor \frac{j}{2} \times \frac{d_n}{d_m} \right\rfloor, \left\lfloor \frac{m - j \bmod 2}{2} \right\rfloor \right)} \left(d_n \times \frac{j}{2} - l \times d_m \right) \right\} \quad (5-11)$$

$$D(T_1 - T_j) = 2 \left\{ \sum_{l=1}^{\left\lfloor \frac{m - j \bmod 2}{2} \right\rfloor} \min \left(d_n \times \frac{j}{2}, l \times d_m \right) \right\} \quad (5-12)$$

当 $m \ll \sqrt{N}$ 时，全能多目标的数据交付开销 C_O 接近等于 $O(n\sqrt{N})$ 。

3. 定向扩散

定向扩散的分析方法与全能多目标相同。为了简化分析，假定定向扩散本地算法构成的树就是由以每个源节点为树根的各棵最短路径树“联合”而成的树。当网络流量载荷轻的时候，这个假设近似有效。当有多条可用最短路径时，定向扩散按照以下规则选择一条最短路径：从一个中心节点到一个源节点，只要沿着对角线链路就能够得到最短路径，那么总是选择对角线链路作为下一个转发跳；否则，即沿着对角线链路不能得到最短路径，选择垂直链

路作为下一个转发跳。这个规则与全能多目标采用的选择规则完全相同。

尽管采用相同的最短路径选择规则，但是定向扩散的数据交付开销 C_d 不等于全能多目标的数据交付开销 C_0 ，其主要原因在于应用层的数据处理。特别是，假如所有源节点发送相同的目标位置估计，并且假定定向扩散可以抑制应用层重复数据，那么定向扩散的数据交付开销是以源节点为树根的所有最短路径树的组合树上的链路数的 2 倍。因此

$$C_d = C(UT_{1 \rightarrow n}) = C(T_1) + \sum_{j=2}^n \left\{ H(T_j - UT_{1 \rightarrow (j-1)}) + D(T_j - UT_{1 \rightarrow (j-1)}) \right\} \quad (5-13)$$

式中

$$H(T_j - UT_{1 \rightarrow (j-1)}) = H(T_j) \quad (5-14)$$

$$D(T_j - UT_{1 \rightarrow (j-1)}) = 2 \left\{ \left[\frac{m + (j \bmod 2)}{2} \right] \times d_n + \sum_{l=1}^{\min \left[\left(\frac{j}{2} \times \frac{d_n}{d_m} \right), \left(\frac{m - j \bmod 2}{2} \right) \right]} \min \left(d_n, d_n \times \frac{j}{2} - l \times d_m \right) \right\} \quad (5-15)$$

类似于 C_0 ，当 $m \ll \sqrt{N}$ 时，定向扩散的数据交付开销 C_d 接近等于 $O(n\sqrt{N})$ 。

4. 比较分析

对于数据交付开销，泛洪的 C_f 比全能多目标的 C_0 高出几个数量级，但是 C_0 仍然大于定向扩散的 C_d ，这是因为 $D(T_1) - D(T_1 - T_j) \geq 0$ 以及 $D(T_j - T_1) \geq D(T_j - UT_{1 \rightarrow (j-1)})$ 。为了验证这个推理，分别绘出参数 N 、 m 、 n 与定向扩散和全能多目标的数据交付开销之间的变化关系曲线，如图 5-10 所示。从图 5-10 (a) 和图 5-10 (b) 中可以看到，随着中心节点、源节点的增多，定向扩散网内处理（比如重复数据抑制）带来的开销优势越来越明显， C_d 的增大速度小于 C_0 。

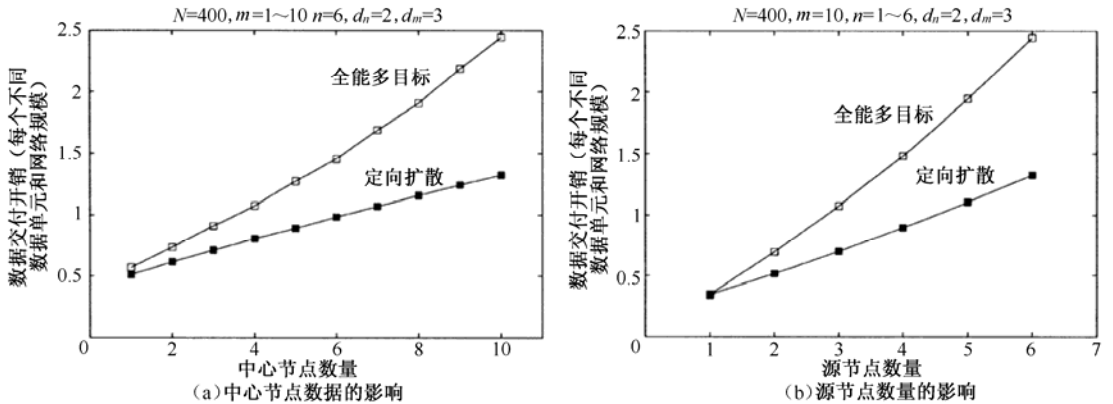


图 5-10 各种参数对定向扩散、全能多目标的影响

5.2.7 定向扩散的仿真评估

采用 ns-2 仿真器实现动物跟踪定向扩散实例。做这种评估研究的目标有五个方面：①验

证和补充定向扩散的分析评估结果；②将定向扩散的性能与理想方案（比如泛洪、全能多目标）放在一起比较，这可以作为定向扩散的直观性正确性检验；③理解动态性（比如节点失效）对定向扩散的影响；④研究无线 MAC 层对定向扩散性能的影响；⑤研究定向扩散对参数选择的敏感性的影响。

1. 性能指标

选择三个性能参数来分析定向扩散的性能并与其他方案比较：

① 平均能耗：表示网络中每个传感器节点的总能耗与各个中心节点所接收到的截然不同事件的数量之比率。平均能耗表示一个传感器节点将一条有用跟踪信息交付给中心节点所做的平均工作，以及表示传感器节点的总寿命时间。

② 平均时延：表示发送一个事件和中心节点接收到该事件之间所观测到的平均单向时延。平均时延定义 WSN 交付的位置估计的时间精度。

③ 不同事件交付率：表示所接收到的不同事件的数量与原始发送事件数量之比率。

将这三个性能参数作为 WSN 规模的函数来研究。

在仿真实验中，假设 WSN 远离过载。因此，传感器节点不会发生拥塞。做这种假设是为了简化对结果的理解。确实存在可靠方法（比如降低网内数据速率或者通过累积主动降低数据质量），用于处理定向扩散 WSN 中的拥塞问题。

尽管重点放在无拥塞工作条件下，但是定向扩散仍然会发生事件丢失问题，特别是在动态条件下。在发生事件丢失的时候，需要使用定向扩散的不同事件交付率。

2. 仿真方法

为了研究定向扩散随着网络规模变化的性能，产生各种大小不同的传感器场。在每个仿真实验中，研究 5 种不同的传感器场，从 50~250 个节点，按 50 个节点递增网络规模。对于 50 节点的传感器场，将 50 个节点随机布置在 160 m×160 m 正方形区域内。每个传感器节点的无线传输距离 40 m。其他规模的传感器场是正方形的适当扩大，传感器节点的无线传输距离保持不变，以便传感器节点的平均密度近似保持不变。这样做的原因是传感器场的连通性是平均节点密度的函数。假如传感器场面积保持不变，但是网络规模增大，那么可以观察到大量网络节点和连通性增强带来的性能结果。在仿真中，进行了网络规模对定向扩散一些机制的影响实验。

ns-2 仿真器实现了一个 1.6 Mb/s 的 IEEE 802.11 MAC 层。仿真中采用经过修改过的 IEEE 802.11 MAC 层。为了较逼真地模拟现实的传感器网络电台（WINS NG 1.0 收发信机）^[7]，修改 ns-2 仿真器的无线能量模型，以便实现空闲期间的功耗约等于 35 ms，约等于其接收功耗（395 mW）的 10%，约等于其发送功耗（660 mW）的 5%。这不是一个完全满意的 MAC 层选择，因为选择 TDMA 类协议，而不是选择基于 RTS/CTS 交互的信道竞争协议作为 WSN 的 MAC 协议是迫于能量效率的缘故。简而言之，这些理由跟电台在空闲期间的能耗有关。采用 TDMA 类 MAC 协议，有可能在空闲期间将电台设置在备用方式。对比之下，IEEE 802.11 电台在空闲期间的能耗约等于发送接收期间的能耗。稍后将分析 MAC 层能量模型的影响，发送侦听的能耗与发送接收的能耗一样多。

最后说明网络载荷。在大多数仿真实验中，采用固定网络载荷：5 个源节点，5 个中心

节点。从传感器场左下方 70 m×70 m 正方形区域内随机选择全部源节点。中心节点均匀分布在传感器场中。每个源节点产生 2 个事件/秒。定向扩散的低数据速率（试探性事件）是每 50 s 1 个事件。事件模拟为 64 B 分组，兴趣模拟为 36 B 分组。按照每 5 s 1 个事件的速率周期性产生兴趣。兴趣持续时间 15 s。否定强化的时间窗口选为 2 s。这些参数的选择由所考虑的特定 WSN（短小事件描述、地理区域内的源节点）及其研究需求来通知。每条性能曲线上的每个数据点是 10 次实验的平均值，可信区间 95%。

3. 比较性评估

第一个仿真实验室比较定向扩散、全能多目标、泛洪的数据分发性能。图 5-11 (a) 表示每个分组的平均能耗与网络节点数的关系曲线。全能多目标方案的每个分组能耗小于泛洪的每个分组能耗的 1/2，这是通过沿着从每个源节点到达每个中心节点的单独路径交付事件来达到这种能量效率的。定向扩散的能量效率比全能多目标方案好得多。对于有些传感器场，定向扩散的能耗只有全能多目标方案的 60%。采用全能多目标方案，由于减少了交付冗余数据的路径数量，因而节省了许多能量。此外，定向扩散极大地受益于网内累积。在动物位置跟踪 WSN 例子中，源节点交付完全相同的位置估计，而中间节点抑制重复的位置估计。比如这就相当于指定区域内存在一只四腿动物。

那么，由于假定 5 个源节点，为什么定向扩散（包含否定强化规则）的能量效率不会比全能多目标方案高出 5 倍左右？第一，两个方案的能耗是可比拟的，发送侦听的能耗不能忽略；第二，选定强化规则和否定强化规则会导致定向扩散频繁地从多条路径上得到高质量数据，由此增加额外的能耗。特别是强行增加一个发送新事件（即至今还未收到过这类事件）的相邻节点的强化规则的主动性是很强的。反之，否定强行增加只连续发送重复事件（即已经收到过这类事件）的相邻节点的否定强化规则是非常保守的。

图 5-11 (b) 表示所观测的平均时延与网络节点数的关系曲线。定向扩散和全能多目标方案的时延大致相当。这是令人可喜的结果。在无拥塞 WSN 中以及没有障碍物条件下，最短路径就是最短时延路径。因此，强化规则似乎是寻找最短时延路径。但是，泛洪的时延几乎比定向扩散、全能多目标方案高出一个数量级。这是 MAC 层的人为现象：为了避免广播碰撞，在全部 MAC 广播上施加了随机时延。泛洪专门采用 MAC 广播，定向扩散只采用广播来传输初始兴趣。对于采用 TDMA 协议作为 MAC 层的传感器电台，泛洪的时延有可能跟定向扩散、全能多目标方案相当。

总之，定向扩散表现出优于全能多目标方案的能耗性能，并且具有良好的时延性能。因为在比较性仿真实验中没有考虑网络动态性并且假定 WSN 无拥塞，所以定向扩散、泛洪、全能多目标全部达到约 100% 的事件交付率。

4. 动态性产生的影响

为了研究动态性对定向扩散的影响，按照如下方法仿真传感器节点失效。对于每个传感器场，反复将一部份（10%或 20%）传感器节点关电停机 30 s。这些节点是从传感器场中均匀选出来的，并且从源节点到达中心节点的最短路径树上同等数量（10%或 20%）的传感器节点也关电停机 30 s。其目的是为了产生定向扩散很可能使用的路径上的节点失效问题以及随机产生 WSN 中其他地方的节点失效问题。此外，不同于前面的仿真实验，每个源节点发

送不同的位置估计（相当于每个源节点“见到”不同的动物）。这样做的原因是在定向扩散过程中抑制来自其他源节点的相同位置估计，因而动态性影响不明显。也可以研究动态性对其他协议的影响，但是，由于全能多目标方案是一个理想方案，没有包括重新计算路由的开销，所以对它们进行比较的意义不是很明显。

动态性仿真实验对数据分发协议施加了很不利的条件：网络中任何时候都有 10% 或 20% 的节点不能使用；此外，不允许在节点失效之间进行任何“时间设置”。即使如此，定向扩散也能够维护合理的事件交付率，如图 5-11（d）所示，同时增加低于 20% 的额外时延如图 5-11（c）所示。而且，在有些情况下，在发生节点失效条件下，平均能耗实际上得到改善。这跟直观感觉有点相反，因为认为定向扩散会消耗一些能量来寻找其他路径。但是，当节点加电工作后，否定强化规则足够保守，从而在正常工作下保持若干条活动的高质量路径。因此，在所设置的动态性强度下，定向扩散不需要作额外的工作。能耗较低的原因在于有些高质量路径发生中断。

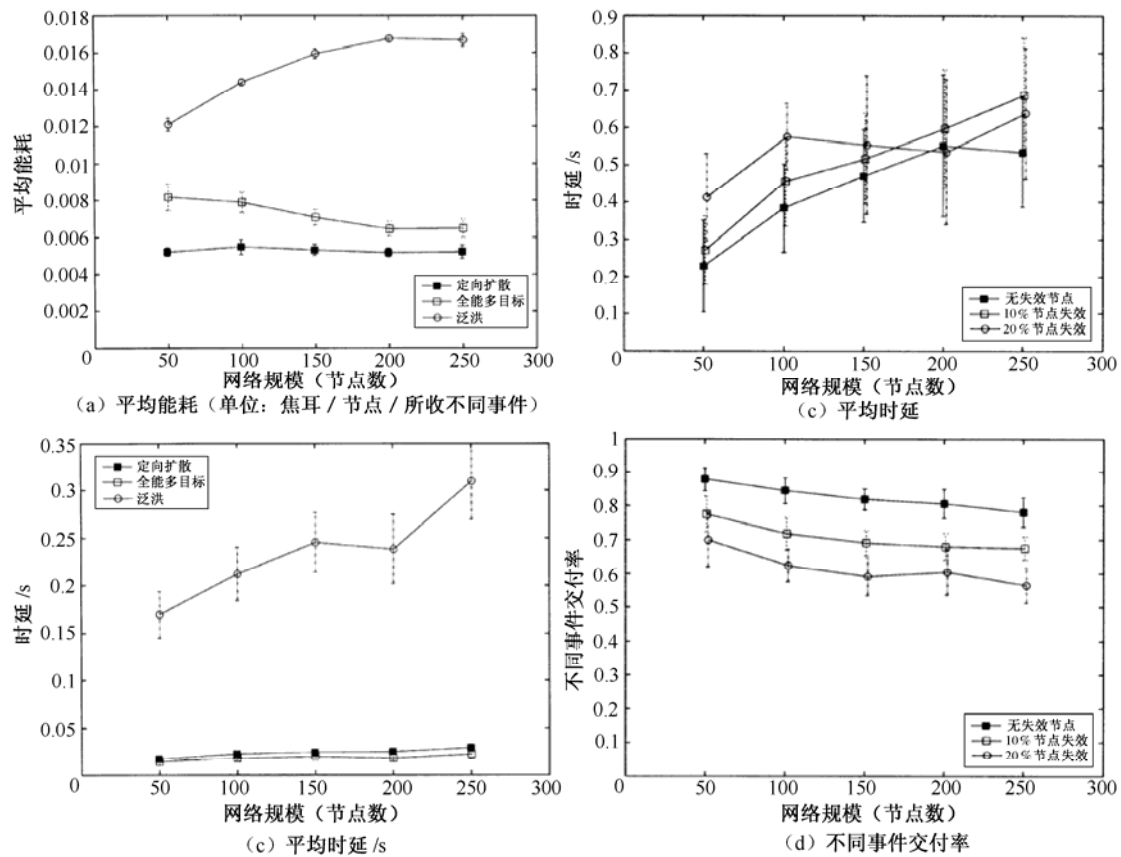


图 5-11 定向扩散的仿真性能

动态性仿真结果表明：在所设置的动态性强度下，定向扩散机制相对稳定。因此，定向扩散在动态条件下不会引起过大的能耗或者过长的交付时延。

参 考 文 献

[1] CLARK,D., AND TENNENHOUSE,D..Architectural Consideration for a New Generation of

- Protocols. In Proc. ACM SIGCOMM(September 1990).
- [2] HEDETNIEMI, S., HEDETNIEMI, S., AND LIESTMAN, A.. A Survey of Gossiping and Broadcasting in Communication Networks. *Networks* 18 (1988).
 - [3] Wendi Rabiner Heinzelman, Joanna Kulik, and Hari Balakrishnan. Adaptive Protocols for Information Dissemination in Wireless Sensor Networks. In *Proceedings of the Fifth Annual ACM/IEEB International Conference on Mobile Computing and Networking (MobiCom'99)*, pp.174-185,1999.
 - [4] Joanna Kulik, Wendi Rabiner Heinzelman, and Hari Balakrishnan. Negotiation-Based Protocols for Disseminating Information in Wireless Sensor Networks. *Wireless Networks* 8,169-185,2002.
 - [5] William Adjie-Winoto, Elliot Schwartz, Hari Balakrishnan, and Jeremy Lilley. The Design and Implementation of an Intentional Naming System. In *Proceedings of the ACM Symposium on Operating Systems Principles*, pages186-201, Charleston, SC, 1999.
 - [6] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. Technical Report 00-732, University of Southern California, March 2000.
 - [7] William J. Kaiser. WINS NG 1.0 Transceiver Power Dissipation Specifications. Sensorla Corp.
 - [8] C. Intanagonwiwat, R. Govindan and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. *Proceedings of the 6th annual international conference on Mobile computing and networking (MobiCom'02)*, pp.56-67, 2000.
 - [9] C. Intanagonwiwat, R. Govindan, D. Estrin, John Heindemann. Directed Diffusion for Wireless Sensor Networking. *IEEE/ACM TRANSACTIONS ON NETWORKING*, Vol.11, No.1, pp.2-16, FEBRUARY 2003.
 - [10] G. Pottle and W. Kaiser. *Wireless Sensor Networks*. Communication8 of the ACM, 2000. To appear.
 - [11] G. Pottle, W. Kaiser, L. Clare, and H. Marcy. *Wireless Integrated Network Sensors*. submitted for publication, 1998.
 - [12] Azzedine Boukerche, Xiuzhen Cheng, Joseph Linus. Energy-Aware Data-Centric Routing in Microsensor Networks. *MSWiM'03*, pp.42-49,2003.
 - [13] D. Estrin, et. al., <http://nesl.ee.ucla.edu/tutorials/mobicom02>
 - [14] M.R. Garey and D.S. Johnson. *Computers and intractability: a guide to the theory of NP-completeness*. Freeman, San Francisco, 1978.
 - [15] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin and D. Ganesan. Building efficient wireless sensor networks with low-level naming. *Proceedings of the eighteenth ACM Symposium on Operating Systems Principles(SOSP'01)*, pp.146-159, 2001.
 - [16] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. *HICSS'00*, 2000.
 - [17] S. Madden, M. J. Franklin and J.M. Hellerstein and W. Hong. TAG: a tiny aggregation service for ad-hoc sensor networks. to appear in *OSDI 2002*.

- [18] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang. Distributed topology control for power efficient operation in multihop wireless ad hoc networks. INFOCOM 2002, Vol.3, pp.1388-1397, 2001.
- [19] K. Akkaya and M. Younis. An Energy-Aware QoS Routing Protocol for Wireless Sensor Networks. in the Proceedings of the IEEE Workshop on Mobile and Wireless Networks (MWN 2003), Providence, Rhode Island, May 2003.
- [20] B. Krishnamachari, D. Estrin, S. Wicker. Modeling Data Centric Routing in Wireless Sensor Networks. in the Proceedings of IEEE INFOCOM, New York, NY, June 2002.
- [21] C. Schurgers and M.B. Srivastava. Energy efficient routing in wireless sensor networks. In the MILCOM Proceedings on Communications for Network-Centric Operations: Creating the Information Force, McLean, VA, 2001.

第 6 章 无线传感器网络分层路由协议

6.1 低能量自适应分群分层 (LEACH)

无线微型传感器网络低能量自适应分群分层 (Low Energy Adaptive Clustering Hierarchy, LEACH) 是一个协议体系, 进行本地计算以减少发送数据量, 采用本地控制进行网络配置和网络操作, 采用 MAC 协议和路由协议进行低能量网络操作。LEACH 将能量高效分群路由协议和 MAC 协议与应用特定数据累积综合在一起, 共同达到良好的系统寿命、时延、应用感觉到的服务质量。LEACH 包含一个能够自组织大量节点的分布式分群技术, 实现所有节点间能量均匀分布的分群自适应算法和群首位置循环算法, 以及节省通信资源的分布式信号处理技术。相对于通用目的的多跳路由法, LEACH 能够将系统寿命提高一个数量级。

6.1.1 LEACH 协议体系结构

LEACH 的设计与开发充分考虑了无线微型传感器网络的独特特性, LEACH 是一个应用特定协议体系。微型 WSN 支持的典型应用是远端环境监视。由于微型 WSN 中各个节点的数据常常是相关的, 所以端用户不需要网络中的所有数据 (冗余数据), 而是需要描述环境中所发生事件的高级数据函数。因为位置相近节点的数据信号的相关性最强, 所以选择采用分群基础设施作为 LEACH 的基础。因此能够对来自分群内部节点的所有数据进行本地处理, 从而减少需要发送给端用户的数据集。特别是, 采用数据累积技术将若干个相关数据信号组合成一个较小的信息集, 这个信息集维持原始信号的有效数据 (即信息内容)。因此, 从分群发送到基站 (Base Station, BS) 的实际数据少得多。

LEACH 对传感器节点和基本网络模型做了一些假设。对传感器节点假定: 所有节点在需要之时具有足够功率将数据发送给 BS, 所有节点能够使用功率控制改变发射功率大小, 每个节点具有支持不同 MAC 协议和完成信号处理功能的计算能力。由于无线硬件技术和低功率计算技术的发展和进步, 这些假设是合理的。对于网络, 假定采用如下网络模型: 节点总是有数据发送给端用户, 相互接近的节点具有相关数据。尽管 LEACH 是在这些假设条件下的优化协议, 但是, 这些假设条件不满足时 LEACH 仍然能够继续工作。本章稍后讨论这些假设条件不满足时 LEACH 的改进方法。

在 LEACH 中, 各个节点自组织成本地群, 每个群有一个节点作为群首。所有非群首节点将其数据发送给自己的群首; 同时群首节点接收所有分群成员节点发送来的数据, 接着对所收数据进行信号处理 (比如数据累积), 然后将其发送给远端 BS。群首节点需要的能量多于非群首节点。假如群首节点首先选择并且在整个系统工作期间固定不变, 那么群首节点将很快耗尽自己的能量。群首节点一旦耗尽自己的能量, 就不能再工作, 其群内所有成员节点失去通信能力。因此, LEACH 采用高能量群首位置随机轮换机制, 让各个传感器节点随机

轮流作为群首节点，以便避免 WSN 中任意传感器节点的电池能量因不停地使用而快速耗尽。这样，作为一个群首节点的能量载荷在各个传感器节点之间均匀分布。

LEACH 操作按照循环重复进行。每个循环从建立阶段开始，即节点自组织成一个一个的群，然后进入稳定状态阶段，即成员节点将其数据发送给其群首节点，然后群首节点将所收成员的数据发送给 BS，如图 6-1 所示。下面将描述群首选择和分布式分群算法以及 LEACH 的稳定状态操作。

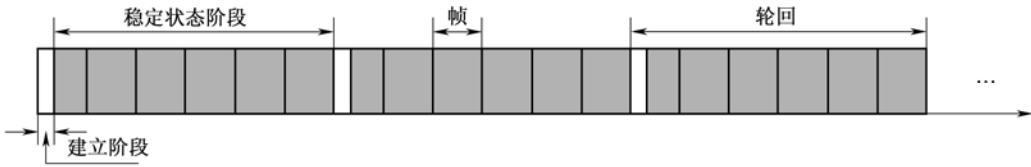


图 6-1 LEACH 操作时序图（在建立阶段构建自适应群，在稳定状态阶段传输数据）

6.1.2 群首选择算法

LEACH 采用分布式算法进行分群，各个传感器节点自行做出决策，不需要任何中心控制。LEACH 的设计目标是：设计一个分群算法，在每个循环中构建一定数量 k 的群；假如开始时各个传感器节点的能量相同，将能量载荷均匀分布在网络中的所有节点之中，以便网络中不存在其能量过度使用而先于其他传感器节点耗尽其能量的节点。由于群首节点需要的能量多于非群首节点，所以要求每个传感器节点轮流充当群首节点。

每个传感器 i 在第 $r+1$ 轮开始时（在时刻 t 启动）以概率 $P_i(t)$ 选择自己作为群首。 $P_i(t)$ 满足本轮群首节点期望个数为 k 。因此，若网络有 N 个传感器节点，则

$$E[\#CH] = \sum_{i=1}^N P_i(t) = k \quad (6-1)$$

确保每个传感器节点每隔 N/k 个循环作为群首的平均次数是相同的。假定 $C_i(t)$ 为节点 i 在最近几个循环 $[r \oplus (N/k) \text{ 个循环}]$ 是否作为群首的指示函数，若节点 i 已经为群首则 $C_i(t)=0$ ，否则 $C_i(t)=1$ ，那么每个节点在第 r 个循环选择以下概率 $P_i(t)$ 成为群首，即

$$P_i(t) = \begin{cases} \frac{k}{N - k \times (r \bmod \frac{N}{k})}, & C_i(t) = 1 \\ 0, & C_i(t) = 0 \end{cases} \quad (6-2)$$

因此，只有满足以下条件的节点才可能在第 $r+1$ 个循环成为群首，最近不是群首并且其可用能量可能多于最近刚刚执行了耗能多功能的节点。

在最初 r 个循环中不是群首的节点期望个数为 $N - kr$ 。经过 N/k 个循环后，所有节点都有一次机会可能成为群首，在随后的循环中全部符合成为群首的条件。若节点 i 在时刻 t 符合成为群首的条件则 $C_i(t)=1$ ，否则 $C_i(t)=0$ ，因此 $\sum_{i=1}^N C_i(t)$ 代表时刻 t 符合群首条件的节点总数，并且

$$E\left[\sum_{i=1}^N C_i(t)\right] = N - k \left(r \bmod \frac{N}{k}\right) \quad (6-3)$$

这就确保每经过 N/k 个循环后，所有节点的能量近似相等。运用式 (6-2) 和式 (6-3)，每个循环的群首节点期望个数等于（稍后介绍最佳 k 的求法）

$$E[\#CH] = \sum_{i=1}^N P_i(t) = \left(N - k \left(r \bmod \frac{N}{k} \right) \right) \frac{k}{N - k \left(r \bmod \frac{N}{k} \right)} = k \quad (6-4)$$

根据如下假设条件选择成为群首的概率：所有节点开始时能量相等，所有节点在每帧均有数据发送。假如各个节点的能量不相等（或者采用事件驱动模型，节点只有在环境中发生某个事件之时才发送数据），那么能量较多的节点应该比能量较少的节点较常成为群首，以确保所有节点大致在相同时刻耗尽能量而停止工作。其实现方法是将节点成为群首的概率设为相对于网络中剩余总能量的节点能量等级的函数，而不是设为节点已经成为群首的次数的纯函数，因此

$$P_i(t) = \min \left\{ \frac{E_i(t)}{E_{\text{total}}(t)} k, 1 \right\} \quad (6-5)$$

式中， $E_i(t)$ 表示节点 i 的当前能量，且

$$E_{\text{total}}(t) = \sum_{i=1}^N E_i(t) \quad (6-6)$$

通过使用这些概率，能量较多节点比能量较少节点更可能成为群首。群首节点的期望个数为

$$E[\#CH] = \sum_{i=1}^N P_i(t) = \left(\frac{E_1(t)}{E_{\text{total}}} + \dots + \frac{E_N(t)}{E_{\text{total}}} \right) k = k \quad (6-7)$$

当节点以相同能量开始时，式 (6-5) 可以用式 (6-2) 近似^[10]。

注意：假如任意节点 i 有 $E_i > E_{\text{total}}/k$ ，其发生概率虽小，却不等于零，那么群首节点的期望个数小于 k 。

为了使用概率式 (6-5)，每个节点必须估计网络中所有节点的总能量。这就需要路由协议，使每个节点确定总能量，而概率[见式 (6-2)]使每个节点完全自行做出决策。避免这个问题的可能方法是网络总的节点能量约等于每个分群节点的平均能量乘以 N 。

注意：计算概率式 (6-2) 和式 (6-5) 要求每个节点已知参数 k 和 N 。事先设置节点的参数 k 和 N 对于动态网络效果不好。最佳分群个数 k 是分散在空间 $M \times M$ 整个区域中的节点数 N 的函数。因此，假定预先确定了 M ，那么节点只需要确定 N 。为此，节点可以给其预先确定的若干跳（约等于 M ）范围内的所有相邻节点发送 hello 消息。每个节点统计其所接收到的 hello 消息个数，这个数值就是 N 的估计值。然后根据这些参数 (N 和 M) 就能够确定所需要的分群个数 k 。这种方法使 LEACH 能够自适应网络的变化，但是其代价是增加了开销。

6.1.3 分群算法

一旦节点使用概率[见式 (6-2) 或式 (6-5)]已经选定自己作为群首后，群首节点必须立即使网络中所有其他节点知道自己在本轮已经作为群首。为此，每个群首节点采用非持续性载波侦听多址访问 (CSMA) MAC 协议广播一条公告消息 (Advertisement Message, ADV)。ADV 是短消息，包含发送节点（群首）的 ID 以及一个分组头，分组头说明本消息是一条公告消息。每个非群首节点按照以下方法确定自己本轮的群首节点：根据群首发送的公告消息的接收信号强度，选择通信能量最低的那个群首作为自己本轮的群首。假定对称传播信道上

纯信号强度，那么与其公告消息接收信号最强的那个群首进行通信所需要的能量最少。注意：通常这个群首就是离传感器最近的那个节点，除非它们之间有障碍物妨碍它们的通信。在相同情形下，随机选择一个群首。

每个节点确定了自己的群首后，必须通知其群首自己是该分群的一个成员。为此，每个节点采用非持续性 CSMA MAC 协议给其群首发送一条加入请求消息 (Joint-REQ)。Joint-REQ 消息也是短消息，包含发送节点及其群首的 ID。

在 LEACH 协议中，群首是本地控制中心，协调其群内的数据传输。群首节点建立起群内的 TDMA 传输时间安排，并将该传输时间安排发送给自己的群内成员。因而确保不会发生数据消息的传输碰撞，并且允许每个非群首节点不在自己的发送时间时一直关闭自己的电台，从而降低了单个传感器的能耗。群内所有节点获悉其群首建立的 TDMA 传输时间安排后，就完成了建立阶段，可以进入稳定状态阶段（数据传输）。LEACH 分群算法的流程如图 6-2 所示。

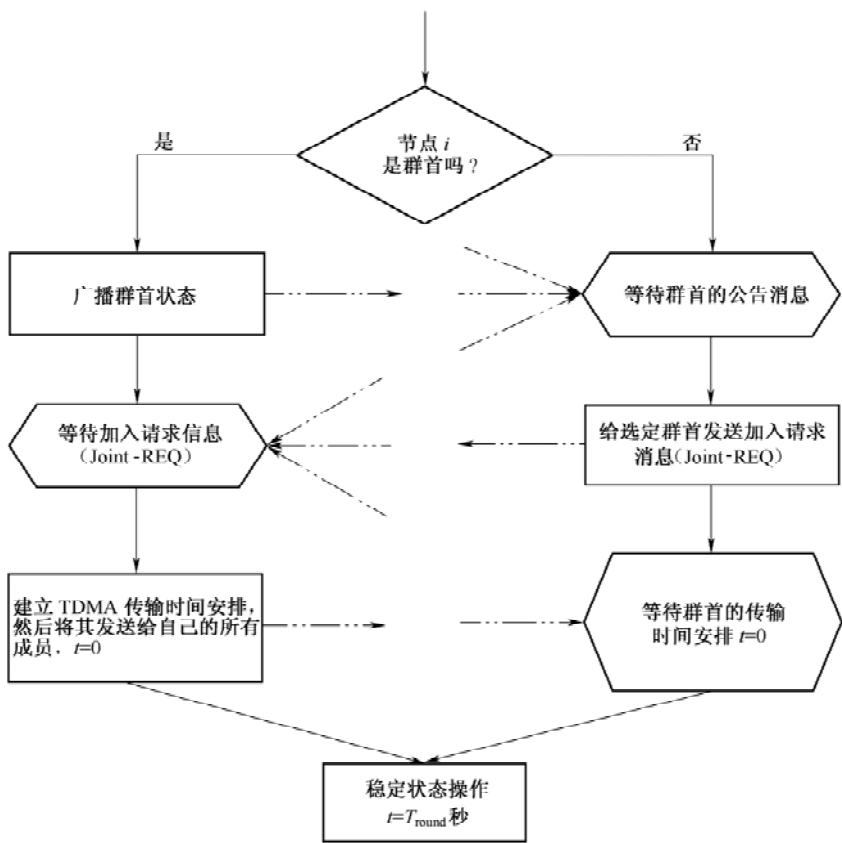


图 6-2 LEACH 分群算法流程图

6.1.4 稳定状态阶段

稳定状态操作按帧进行，成员节点在其分得的发送时隙内将其数据发送给自己的群首节点，但是在一个时隙内最多只能发送一个分组。每个时隙的长度固定不变，所以发送一个数

据帧所需要的时间跟群内节点数量有关。图 6-3 (a) 表示一个 LEACH 循环的时序图。假定所有节点是同步的、在相同时刻启动建立阶段。例如，BS 给节点发送同步脉冲，就可实现全部节点的同步。

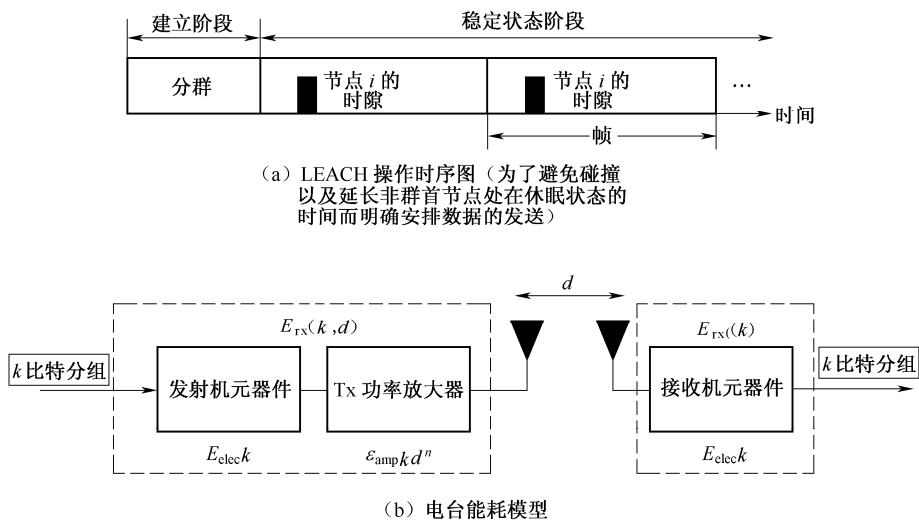


图 6-3 LEACH 操作时序、电台能耗模型

为了降低功耗，每个非群首节点根据所收群首的公告消息的信号强度，采用功率控制设置发送功率。为了确保动态环境下的连接，节点可以将其发送功率设为稍高于到达群首所需要的最低功率，或者群首可以给其每个成员节点发送一个短消息，以便通知成员节点提高或者降低发射功率。此外，关闭每个非群首节点的电台，直到到达该节点的发送时间才重新打开其电台。因为针对所有节点有数据发送给群首作了设计优化，所以采用 TDMA 传输时间安排能够高效利用带宽，是低时延、能量高效实现法。

群首必须处于工作状态，以便接收其成员节点发送来的全部数据。群首一旦接收到所有数据，就立即对数据进行累积、组合处理，以便加强公共信号、削弱信号间的不相关噪声。在后面的分析中假定完全相关，以便能够将所有单个信号组合成一个典型信号。群首将最后得到的数据发送给 BS。由于 BS 可能比较远以及数据消息很大，所以这是一种高能量传输。

前面的讨论描述了群内的通信，其 MAC 协议和路由协议要确保群内成员节点低能耗和群内数据消息无碰撞。但是，无线信道本质上是一种广播媒介。同样地，一个群内的发送将影响（并且常常恶化）附近某个群的通信。为了降低群间干扰，在 LEACH 中，每个分群采用直接序列扩频（Direct Sequence Spread Spectrum, DSSS）进行通信。每个分群采用一个独特扩频码，群内所有成员节点采用该扩频码将其数据发送给群首，群首采用该扩频码对所有所收能量进行滤波，叫做基于发射方的码分，这是因为群内所有发射机都采用同一个扩频码。预先制作好一张扩频码表，那么第一个发送其位置的群首分得表中的第一个扩频码，第二个发送其位置的群首分得表中的第二个扩频码，以此类推。假如分群的个数大于扩频码的数量，那么有些分群将使用相同的扩频码，假如这些分群相距较近，那么可能引起数据传输碰撞。假如有足够多的扩频码，那么在去相关的时候，相邻分群发送的信号将作为噪声被过滤掉，因此不会干扰群内节点的发送。为了降低干扰邻近分群的概率以及降低自己的功耗，每个节点调整自己的发射功率。因此，相互重叠的传输很少，为确保低碰撞概率，仍然很少采用数

据扩散法。

采用固定扩频码和 CSMA 将数据从群首发送到 BS。群首有数据发送时，必须首先侦听信道，确定是否有其他节点正在使用 BS 扩频码发送：假如有，则等待一段时间再准备发送；否则，使用 BS 扩频码发送其数据。

也可以采用其他信道技术，比如每个分群采用不同的频带（如 FDMA）。但是，由于 LEACH 的分群数量不是固定不变的，所以在分群较少时采用 DSSS 能够确保所有节点接收较好的通信信号。按照某种固定信道方案动态分配频带从而能够利用整个带宽是比较困难的。当然，采用 DSSS 的缺点是需要严格的时间同步，从而群首和非群首节点之间可能必须增加新的通信开销。

6.1.5 LEACH-C: BS建立分群

LEACH 分布式分群算法尽管有优势，但是却不能保证群首节点的布置和（或者）数量。由于分群是自适应的，所以在一个给定循环期间获得一个欠妥的分群结构不会对总体性能造成太大影响。但是，采用中心控制算法将群首节点分散在整个网络中，有可能得到更好的分群结构。这就是 LEACH-中心（LEACH-Centralized, LEACH-C）版的基础，LEACH-C 采用中心分群算法，其稳定状态协议与 LEACH 相同。

在 LEACH-C 的建立阶段，每个节点将其当前位置（可以通过 GPS 接收机来确定）信息和能量等级发送给 BS。BS 除了确定良好的分群结构，还要确保能量载荷均匀分布在所有节点之间。为此，BS 计算平均节点能量，其能量低于该平均能量的任何节点在本轮不能作为群首。为了使用剩余的节点作为群首，BS 采用仿真强化算法^[16]来解决寻找 k 个最佳群首的难以解决问题（NP-Hard），从而建立分群。该算法使所有非群首节点与其群首之间的距离平方之总和最小，从而使非群首节点给其群首发送数据的能耗最小。通信能量常常是不能采用距离精确测量的。但是，采集所有节点之间的通信信道信息是不切实际的。因此，采用根据 GPS 坐标计算得到的距离是通信所需能量的近似值。

一旦找到群首和建立起相关分群，那么 BS 广播一条消息，该消息包含每个节点的群首 ID。假如一个节点的群首 ID 与自己的 ID 匹配，那么该节点就是群首；否则，该节点确定自己的 TDMA 数据发送时隙，然后进入休眠状态，直到到达其数据发送时刻为止。LEACH-C 的稳定状态协议与 LEACH 完全相同。

6.1.6 LEACH的分析与仿真

对于即使几十个节点的适度大小的网络，分析模拟所有节点之间的交互也是非常困难的。因此使用网络仿真器 ns-2 来评估 LEACH 以及与其他协议进行比较。比较 LEACH 与 LEACH-C、MTE 路由、固定分群的系统寿命、能耗、数据传输量、时延。

采用最低发射能量（Minimum Transmission Energy, MTE）路由协议选择中间节点，要求平方距离之和[因此，在假定 d^2 功率损耗条件下，总发射能量 $E_{tx}(d)$]最小。因此，对于三个节点 A、B、C，节点 A 当且仅当下述条件成立时才能通过节点 B 将其发送传输给节点 C

$$E_{tx}(d=d_{AB}) + E_{tx}(d=d_{BC}) < E_{tx}(d=d_{AC})$$

或者 $d_{AB}^2 + d_{BC}^2 < d_{AC}^2$ 。这种方法没有考虑电台数据发送和接收的能耗，因此可能不能真正选

出最低能量路由。

对于 MTE 路由，每个节点运行启动程序，以便确定其下一跳相邻节点，下一跳相邻节点定义为朝 BS 方向的最近节点。假定每个节点知道网络中所有节点的位置，以便简化 MTE 路由的建立过程。一般地，需要某种初始化过程将位置信息分发到整个网络中。数据分组沿着下一跳相邻节点传递，直到到达 BS 为止。由于 MTE 路由没有中心控制，建立固定 MAC 协议（如 TDMA）非常困难，所以每个节点在发送数据前采用 CSMA 侦听信道。假如信道侦听为忙，那么节点退避；否则，节点将其数据发送到下一跳节点。当节点能量耗尽时，重新计算路由，以便确保与 BS 的连接。在仿真中，没有说明这种路由更新的能量要求或者时延要求。每个节点每隔 t_{delay} 秒发送自己的数据，设置 t_{delay} 时要求拥塞最轻但要确保信道带宽的高效使用。若 t_{delay} 太小，节点则会在早先数据集能够到达 BS 之前结束其数据发送，队列将变长，将发生严重的数据碰撞，BS 接收到的信息将非常少。若 t_{delay} 太大，当信道可以用来发送数据时信道却处于空闲状态。根据以下三个参数设置 t_{delay} ：①网络中的总节点数 N ；②消息到达 BS 的平均跳数；③消息通过一跳传递所需要的时间。

对于固定分群，开始 BS 采用与 LEACH-C 相同的方法将节点组织成一个一个的分群，以便确保获得良好的分群结构。这些分群和群首在整个网络有效期间保持不变。像 LEACH 和 LEACH-C 那样，节点在每个数据传输帧将其数据发送给自己的群首（采用 TDMA 和 DSSS 扩频码确保群间干扰最小），群首累积其成员节点的数据，然后将最后得到的数据发送给 BS。当群首节点耗尽其能量时，其成员节点失去与 BS 的通信能力，必然“死掉”（即失效）。

1. 实验建立

在仿真实验中，采用 100 个节点的网络，节点随机分布在 $(x=0, y=0)$ 和 $(x=100, y=100)$ 之间，BS 位于 $(x=50, y=175)$ 。信道带宽设为 1 Mb/s，每条数据消息长 500 B，各种分组的分组头均长 25 B。

假定一个简单的电台硬件能耗模型：发射机的能耗在电台元器件和功率放大器上，接收机的能耗在电台元器件上，见图 6-3（b）。对于前述实验，采用自由空间（ d^2 功耗）信道模型和多径衰落（ d^4 功耗）信道模型，这两个模型依赖发射机和接收机之间的距离 d 。注意这是一个简化模型；一般地，无线电波传播是高度变化的，难以模拟。采用功率控制适当设置功率放大器，以便转化这些功耗，假如距离 d 小于门限值 d_0 ，则采用自由空间（ f_s ）模型；否则，采用多径（MP）衰落模型。因此，在距离 d 发送一条长度 l 比特消息的电台耗能为

$$E_{\text{tx}}(l, d) = E_{\text{tx-elec}}(l) + E_{\text{tx-amp}}(l, d) = \begin{cases} lE_{\text{elec}} + le_{\text{fs}}d^2, & d < d_0 \\ lE_{\text{elec}} + le_{\text{mp}}d^4, & d \geq d_0 \end{cases} \quad (6-8)$$

接收这条消息的电台耗能为

$$E_{\text{rx}}(l) = E_{\text{rs-elec}}(l) = lE_{\text{elec}} \quad (6-9)$$

电台电子元器件耗能 E_{elec} 依赖诸如数字编码、调制、滤波、信号扩频之类的因素，功率放大器耗能 $e_{\text{fs}}d^2$ 或者 $e_{\text{MP}}d^4$ 依赖到达接收机的距离以及可接受的比特误码率。对于这里描述的仿真实验，通信能量参数设置如下： $E_{\text{elec}}=50 \text{ nJ/bit}$ ， $e_{\text{fs}}=10 \text{ pJ/bit/m}^2$ ， $e_{\text{MP}}=0.0013 \text{ pJ/bit/m}^4$ 。数据累积耗能设为 $E_{\text{DA}}=5 \text{ nJ/bit/signal}$ 。

2. 分群的最佳个数

在 LEACH 中, 创建分群算法, 以确保每轮的分群期望个数 k 。可以采用计算和通信能量模型, 通过分析确定 LEACH 的系统参数 k 的最佳值。假定 N 个节点均匀分布在一个 $M \times M$ 区域中。假如有 k 个分群, 每个分群平均 N/k 个节点[一个群首加上 $(N/k)-1$ 个非群首节点]。每个群首的能耗用于接收节点发送的信号、累积所收信号、将累积信号发送给 BS。由于 BS 离节点较远, 所以其能耗大致符合多径衰落模型 (d^4 功耗)。因此, 群首节点在一个帧中的耗能为

$$E_{CH} = lE_{\text{elec}} \left[\left(\frac{N}{k} \right) - 1 \right] + lE_{\text{DA}} \left(\frac{N}{k} \right) + lE_{\text{elec}} + le_{\text{MP}} d_{\text{toBS}}^4 \quad (6-10)$$

式中, l 表示每条数据消息的比特数, d_{toBS} 表示群首节点到达 BS 的距离, 假定数据完全累积。

每个非群首节点在每帧中只需给其群首进行一次其数据发送。每个非群首节点到达其群首节点的距离大概比较短, 所以其能耗符合 Friss 自由空间模型 (d^2 功耗)。因此, 每个非群首节点的耗能为

$$E_{\text{non-CH}} = lE_{\text{elec}} + le_{\text{fs}} d_{\text{toCH}}^2 \quad (6-11)$$

式中, d_{toCH} 表示该节点 (非群首节点) 到达其群首节点的距离。每个分群占领的区域大致等于 M^2/k 。一般地, 每个分群所在区域是任意形状的区域, 节点分布为 $p(x,y)$ 。从节点 (非群首节点) 到达其群首节点 (假定处在分群面积的中心位置) 的平方距离期望值为

$$E[d_{\text{toCH}}^2] = \iint (x^2 + y^2) p(x,y) dx dy = \iint r^2 p(r,\theta) r dr d\theta \quad (6-12)$$

假定该区域是一个半径 $R = (M/\sqrt{\pi k})$ 的圆, $p(r,\theta)$ 对于 r 和 θ 是恒定的, 那么式 (6-12) 可以简化为

$$E[d_{\text{toCH}}^2] = p \int_0^{2\pi} \int_0^{M/\sqrt{\pi k}} r^3 dr d\theta = \frac{p}{2\pi} \frac{M^4}{k^2} \quad (6-13)$$

假如节点密度在整个分群区域中是均匀的, 那么 $p = [1/(M^2/k)]$, 以及

$$E[d_{\text{toCH}}^2] = \frac{1}{2\pi} \frac{M^2}{k} \quad (6-14)$$

因此, 在这种情况下有

$$E_{\text{non-CH}} = lE_{\text{elec}} + le_{\text{fs}} \frac{1}{2\pi} \frac{M^2}{k} \quad (6-15)$$

一个分群在本帧中的能耗为

$$E_{\text{cluster}} = E_{CH} + \left(\frac{N}{k} - 1 \right) E_{\text{non-CH}} \approx E_{CH} + \frac{N}{k} E_{\text{non-CH}} \quad (6-16)$$

本帧的总耗能为

$$E_{\text{total}} = kE_{\text{cluster}} = l \left(E_{\text{elec}} N + E_{\text{DA}} N + ke_{\text{MP}} d_{\text{toBS}}^4 + E_{\text{elec}} N + e_{\text{fs}} \frac{1}{2\pi} \frac{M^2}{k} N \right) \quad (6-17)$$

设 E_{total} 对 k 的导数为零, 则可得到分群个数的最佳期望值

$$k_{\text{opt}} = \frac{\sqrt{N}}{\sqrt{2\pi}} \sqrt{\frac{e_{\text{fs}}}{e_{\text{mp}}} \frac{M}{d_{\text{toBS}}^2}} \quad (6-18)$$

对于仿真实验参数配置: $N=100$ 个节点, $M=100$ m, $e_{\text{fs}}=10$ pJ, $e_{\text{MP}}=0.0013$ pJ,

$75\text{ m} < d_{\text{toBS}} < 185\text{ m}$, 分群个数的最佳期望值 $1 < k_{\text{opt}} < 6$ 。

100 节点网络的仿真结果验证了这些分析结果。

3. 能量增益

在能量仿真实验中, 每个节点开始只有 2 J 能量, 且发送给 BS 的数据没有限制, 在每个循环开始采用概率[见式 (6-2)]确定其群首的状态。每个循环持续 20 s。要求每个循环的持续时间保证平均每个节点在整个仿真期间承担一次群首和若干次非群首。跟踪数据分组转发到 BS 的速率以及获取给 BS 的数据所需要的能量。节点在仿真期间耗尽其有限能量后, 就不再发送和接收数据。

对于这些仿真实验, 节点只要发送数据、或者接收数据、或者进行数据累积处理, 就消耗能量。采用扩频提高发送比特数量, 因此增加了电台元器件的耗能。既没有假定任何静态能耗, 也没有假定载波侦听操作的能耗, 因此这里得到的结果没有考虑在 LEACH 中采用 TDMA 相对于在 MTE 中采用 CSMA 可能带来的能量好处。

尽管质量是应用特定和依赖数据的质量, 但是是一种不依赖应用的质量测试方法是测量 BS 接收的数据量 (实际数据信号量或者一个累积信号代表的数据信号量)。BS 接收的数据越多, 那么 BS 对远端环境的观测就越精确。假如群内所有节点观测同一个事件, 那么实际有效数据包含相同信息, 发送有效数据或者累积数据不会导致质量下降。另一方面, 假如群内各个节点观测不同事件, 那么群首选出最强事件 (群内各个成员信号中的最强信号), 并将其作为本群的数据而发送给 BS。在这种情况下, 将各个信号累积成一个代表信号将导致质量下降。由于采用无线电波传播, 无线信号传播依赖诸如信号特性、信号源与传感器之间的距离、传感器灵敏度之类的因素, 所以很难对信号传播进行量化。假如群内节点间的距离相对于离所能够观测事件的距离较短, 或者假如环境中发生的事件间的距离很远, 那么各个节点观测同一个事件的概率很高。在仿真实验中假定一个群内所有节点观测同一个事件。

图 6-4 (a) 表示 BS 随着时间的推进而接收的数据信号总量 (对于 MTE 是实际数据, 对于 LEACH、LEACH-C、固定分群是有效数据) 以及 BS 在给定能量下所接收的数据总量。图 6-4 (a) 表明: LEACH 在仿真中发送给 BS 的数据多于 MTE 路由。MTE 需要较多时间将来自节点的数据发送给 BS 的原因在于每条消息需要经过多跳传输。在其他协议中, 每条消息经过一跳传输到达群首, 群首再对其进行累积处理。将累积信号发送给 BS 大幅度减少了所发送的数据。图 6-4 (b) 表示 BS 在给定能量下所接收的数据总量。图 6-4 (b) 表明: LEACH 和 LEACH-C 单位能量交付的数据最多, 同时达到能量效率和时延效率。诸如 MTE 之类的路由协议没有进行本地计算, 因而发送给 BS 的数据量没有得到减少。

图 6-4 (a) 和图 6-4 (b) 表明 LEACH 的效率没有 LEACH-C 高 (LEACH-C 单位能量交付的数据比 LEACH 多 40%)。这是因为 BS 具有网络中所有节点的位置信息和能量信息, 所以 BS 建立分群需要较少的数据传输能量, 并且能够得到更好的分群结构。此外, BS 分群算法确保每个循环操作 $k=5$ 个分群。由于仿真中只有 100 个节点, 尽管 LEACH 每个循环操作的分群期望个数 $k=5$, 但是并不是每个循环操作总是有 5 个分群。

图 6-4 (c) 给出活动节点总数与仿真时间的关系。MTE 节点活动时间保持较长, 这是因为发送给 BS 的数据少得多。绘出 BS 每接收一个数据而仍然保持活动的节点总数[见图 6-4 (d)], 从中可以看到: 在相同数量节点失效条件下, LEACH 交付的有效数据比 MTE 路由多 10 倍。MTE 发送数据给 BS 需要更多能量的理由有两个 (因此交付相同数量数据将导致较多节点失

效)：碰撞和无数据累积。因为 MTE 没有中心控制单元来管理和控制节点的分组发送时间和接收时间，所以碰撞导致成功发送每条消息所需能量的增加。此外，MTE 中的每条消息必须传递通过约 $0.6\sqrt{N} = 6$ 跳才能够到达 BS。寻找平均传递跳数的分析类似于寻找 $E[d_{toCH}^2]$ 的分析，请参阅第 5 章定向扩散分发协议。而 LEACH 中的每条消息只需要传递一跳就能够到达 BS，这是因为 LEACH 在群首节点采取了数据累积处理。当然，这里假定 LEACH 采取完全累积；当放宽这个假设时，采用 MTE 的优势就非常大。

图 6-4 (c) 和图 6-4 (d) 说明了固定分群的性能表现差的原因，如同从图 6-4 (a) 和图 6-4 (b) 中看到的結果一样：群首节点迅速失效，导致其全部成员节点失效。因此，使网络中每个节点轮流承担群首使得 LEACH 能够实现长于固定分群的寿命。

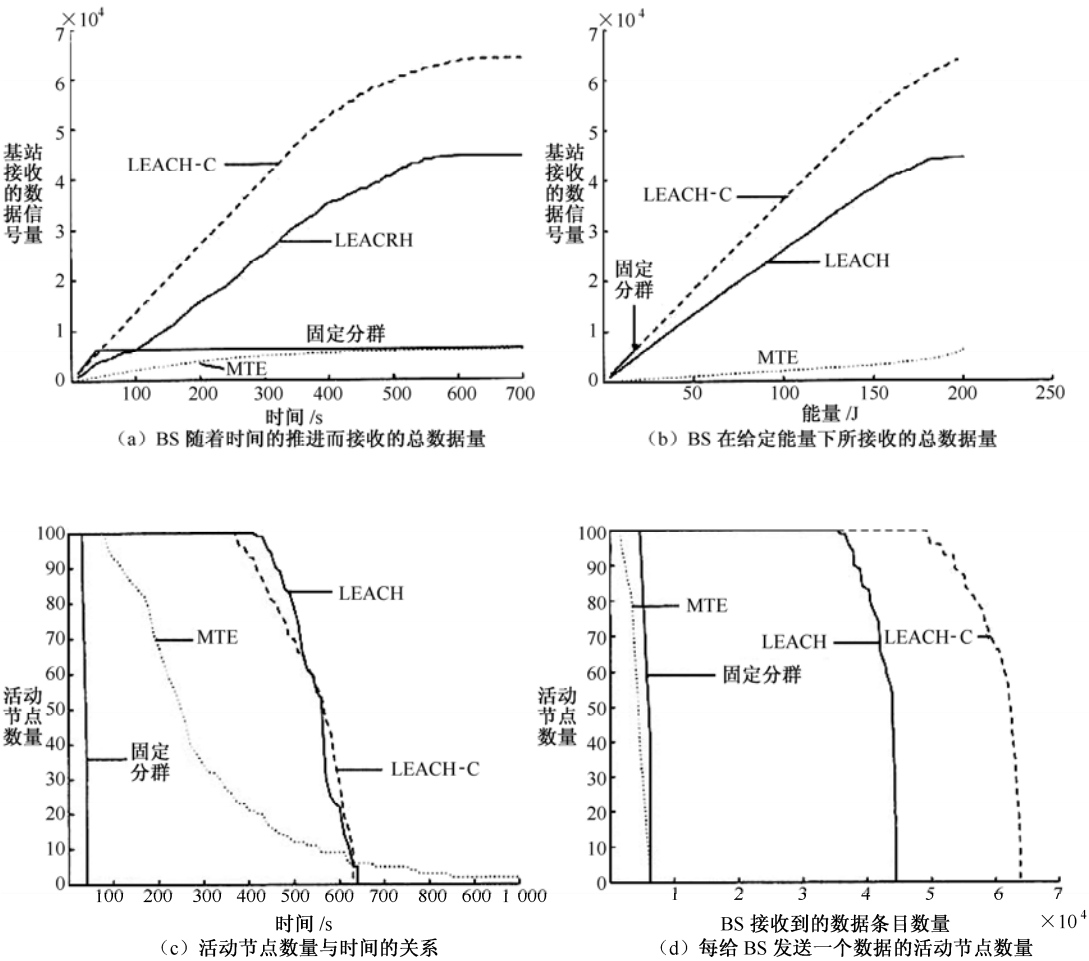


图 6-4 LEACH 的性能

6.2 两层数据分发协议 (TTDD)

两层数据分发 (Two-Tier Data Dissemination, TTDD) 协议用于解决大规模 WSN 潜在的多个数据源节点对多个移动中心节点的可扩展的高效数据分发问题。数据源节点就是产生感知

数据而需要报告有关激励因素的传感器节点，激励因素是一个目标或者感兴趣的一个事件。中心节点就是从 WSN 中收集这些数据报告的用户。激励因素和中心节点的数量可能随着时间的推进而变化。比如在图 6-5 中，一队士兵通过布置在战场中的 WSN 收集坦克运动信息。坦克周围的传感器检测坦克，共同协作累积数据，其中一个传感器节点产生数据报告。士兵收集这些数据报告。这里假定只考虑由静止传感器节点组成的网络，但是中心节点可以动态改变其位置。在上述例子中，作为中心节点的士兵可以到处走动，但是必须能够连续接收数据报告。

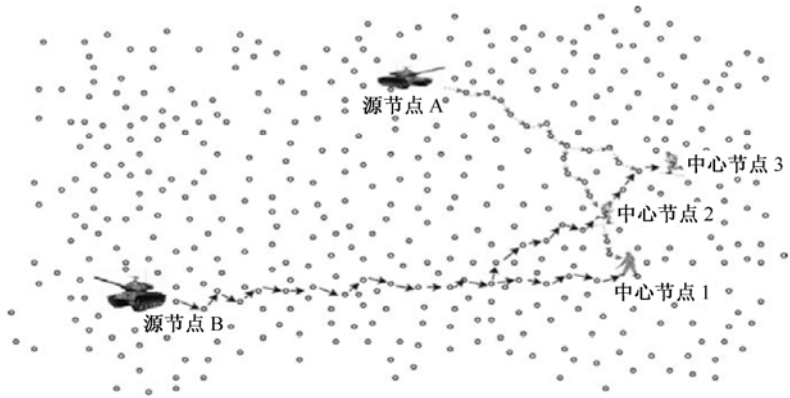


图 6-5 一个 WSN 例子：士兵使用 WSN 检测坦克的位置

中心节点移动给大规模 WSN 的数据分发带来了新的挑战。定向扩散假定每个中心节点需要连续将其位置信息广播到整个传感器场中，所以所有传感器节点都得到通知其随后数据报告的发送方向。但是，多个中心节点频繁的位置更新不仅加重了无线传输的碰撞，而且导致迅速消耗传感器节点有限的电池能量。现有的数据分发方法不能对这个问题提供可扩展的高效解决方法。

TTDD 不是从每个中心节点开始将查询消息发送给所有传感器节点来实现数据转发信息的更新，而是采用栅格结构，只有栅格点上的传感器节点才需要获取转发信息。检测到一个激励因素后，不是被动等待中心节点的数据查询（许多数据分发协议采用这种方法），而是数据源节点主动构建一个覆盖整个传感器场的栅格结构，在离栅格点最近的传感器节点上建立转发信息（下面将其称为分发节点）。采用栅格结构后，从中心节点发出的查询消息通过两层传递到达源节点。低层在中心节点当前位置的本地栅格正方形内（因此称为蜂窝），高层由栅格上的分发节点组成。中心节点在蜂窝内泛洪查询消息。最近分发节点接收到所申请数据的查询消息后，将其转发给朝源节点方向的上行分发节点，上行节点又进一步向上转发，直到查询消息到达源节点或者到达已接收到源节点发送数据的数据分发节点（比如接收到其他中心节点的查询消息后）为止。这个查询转发过程提供了到达中心节点的路径信息，确保源节点的数据沿着查询消息转发路径的相反方向从源节点开始通过两层传递到达中心节点。

TTDD 设计采用如下论据：传感器节点既是静止的，又具有位置意识。因为假定传感器节点知道自己的位置以便于对感知数据加标签，又因为传感器节点位置是静止的，所以 TTDD 能够使用简单的贪婪地理转发路由来建立和维护栅格结构，并且开销低。对每个数据源节点采用栅格结构，从多个中心节点发出的查询消息被限制在其本地蜂窝内，因此避免了多个中心节点全网泛洪的过度能耗和网络开销。中心节点在大于其蜂窝范围移动而离开原来位置时，重新进行本地数据查询泛洪，查询消息将传递到达新的分发节点。沿着源节点方向传递，查

询消息最终被某个已经接收到源节点发送数据的分发节点所接收而不再进一步转发。然后分发节点沿着查询消息传递路径的反方向向下朝中心节点转发数据。这样，即使中心节点连续移动，高层数据转发递增式变化，中心节点仍然能够连续接收到数据。而且由于栅格点上的传感器节点（作为分发节点）参与数据分发，所以其他传感器节点不需要维护状态。因此，TTDD 能够扩展到大量的源节点和中心节点。

6.2.1 两层数据分发

TTDD 的基本设计采用如下网络设置：

① 大量同类传感器节点覆盖一个巨大传感器场，传感器节点采用短距离电台进行通信。通过多跳数据转发实现远距离数据交付。

② 每个传感器节点知道自己的位置（比如通过接收 GPS 信号或者其他定位技术获取位置）。但是，移动中心节点可能知道，也可能不知道自己的位置。

③ 一旦出现一个激励因素，那么其周围的传感器节点立即共同处理信号，其中一个传感器节点成为产生数据报告的源节点。

④ 中心节点（用户）查询网络，收集感知数据。传感器场中可能存在多个移动中心节点，中心节点的数量随着时间的推进而变化。

上述假设与所建立的真实传感器（比如 UCLA 的 WINS NG 节点^[15]、SCADDS PC/104^[4]、加州大学伯克利分校的 Mote^[10]）模型一致。

此外，TTDD 设计假定传感器节点知道自己承担的任务（比如所观察的每种可能的激励因素的信号形式）。每种任务代表 WSN 的一种感知任务。在图 6-5 中，WSN 的任务是收集和返回坦克的当前位置。当 WSN 任务可能偶尔改变时，可以将新任务泛洪到整个传感器场中的所有传感器节点。假定一个 WSN 的任务极少发生变化，因此相对于感知数据交付开销，任务分发开销忽略不计。

一个源节点只要产生数据，就立即构建一个栅格结构，准备数据分发。源节点开始时将自己的位置作为栅格上的一个交叉点，给其四个相邻交叉点发送数据通知消息。每条数据通知消息最终被其所指定的最近交叉点的一个传感器节点所接收，该传感器节点存储源节点信息，然后将该消息转发给相邻交叉点（不包括将该消息发送来的那个相邻交叉点）。这种数据通知消息的递进式传播通知交叉点最近的所有传感器节点成为给定源节点的分发节点。

一旦建立起特定源节点的栅格，那么中心节点可以立即在其蜂窝内泛洪查询消息，以便接收数据。栅格上最近分发节点接收到查询消息后，朝源节点方向将其转发给上行分发节点。所请求的数据沿着反方向朝中心节点下行传递。

简单的 TTDD 操作提出了几个研究挑战。例如，假定传感器的位置是随机的且没必要处在栅格的交叉点上，那么栅格点附近的传感器节点如何确定哪个作为分发节点？一旦数据流开始传递，那么如何随着中心节点移动而确保连续数据交付？假定单个传感器节点经常遇到意外故障，那么栅格结构建立完之后如何维护？下面详细介绍这些问题的解决方法。

6.2.2 栅格结构

为了简化陈述，考虑二维传感器场。一个源节点将二维传感器场分成蜂窝栅格。每个蜂

窝为 $\alpha \times \alpha$ 的正方形。源节点本身处在栅格的某个交叉点上，广播数据通知消息，使数据通知消息到达栅格上所有其他交叉点（将这些交叉点叫做分发点）。对于位置 $L_s=(x,y)$ 上的一个特定源节点，分发点的位置 $L_p=(x_i,y_j)$ 满足：

$$\{x_i=x+i\alpha, \quad y_j=y+j\alpha, i, j=\pm 0, \pm 1, \pm 2, \pm 3, \dots\}$$

假定已知源节点位置 (x, y) 及蜂窝大小 α ，源节点计算其周围四个相邻分发点的位置。源节点采用简单的贪婪地理转发路由对每个分发点 L_p 发送一条数据通知消息，即将数据通知消息转发给离 L_p 最近的相邻节点。同样地，相邻节点又继续转发数据通知消息，直到 L_p 的一个较近节点（而不是 L_p 的所有相邻节点）接收到数据通知消息为止。假如这个节点与 L_p 间的距离小于门限 $\alpha/2$ ，那么这个节点成为源节点的一个分发节点，对分发点 L_p 服务。假如数据通知消息被一个节点所接收，该节点到达指定分发点的距离大于 $\alpha/2$ ，那么该节点只需丢掉数据通知消息。

分发节点为栅格结构存储若干信息，包括数据通知消息、对其服务的分发点 L_p 、上行分发节点的位置。然后分发节点又进一步将数据通知消息转发给自己的相邻分发点（不包括将数据通知消息发送来的那个上行相邻节点）。数据通知消息递进式地传递到整个传感器场中，因此栅格上的每个分发点都会得到一个分发节点的服务。通过数据通知消息中的序列号识别来自不同相邻分发点的重复的数据通知消息，只需丢掉重复的数据通知消息。

1. 栅格结构说明

由于上述栅格建立过程没有假定事先知道中心节点的位置，所以建立均匀栅格，所有分发点等距离 α 均匀分布在栅格上，以便尽可能均匀地分发数据通知消息。任何传感器节点不需要知道全网拓扑，每个传感器节点只根据其本地相邻节点的信息进行操作。

在 TTDD 中，分发点作为选择分发节点的参考位置。选择的分发节点尽可能接近分发点，因此分发节点形成一个几乎均匀的栅格基础设施。但是，不要求分发节点完全是分发点的最近节点。严格地说，由于拓扑的不规则性，TTDD 确保分发节点离分发点是本地最近的，但不必是完全最近的。这不影响 TTDD 的正确操作。理由是：每个分发节点将自己的位置（不是分发点的位置）信息写入到将转发的数据通知消息中。这样下行分发节点就能够将随后的查询消息转发到该分发节点，尽管这个分发节点在真实栅格上不是离分发点完全最近的分发节点。

当一个分发点落入没有传感器节点的空区域时，可能在空区域边界上停止发送数据通知消息。但是因为每个分发节点将数据通知消息转发给所有另外三个分发点，所以能够沿着另外一条路径绕过空区域继续传递数据通知消息。只要栅格不发生分割，就可以采用其他路径绕过空区域继续传递数据通知消息。

在每个源节点基础上建立栅格，不同的源节点得到不同的分发节点集。这种设计选择提高了可扩展性，提供载荷平衡和更好的强壮性。当有多个源节点时，只要各个源节点的栅格不重叠，那么一个分发节点只需要一个或者几个源节点的状态，从而允许 TTDD 扩展到大量源节点情形。按照每个源节点构建栅格能够有效地将数据分发载荷分散到不同的传感器节点上，避免发生瓶颈。其根据在于每个传感器节点的能量有限，其电台传输带宽也有限。按照每个源节点构建栅格能够提高系统面对节点失效的强壮性。

栅格大小 α 是一个关键参数。设置蜂窝大小的一般原则是将中心节点移动的影响限制在单个蜂窝内（中心节点移动影响局部化），以便高层栅格转发保持稳定。 α 的选择影响能量

效率和状态复杂性。

6.2.3 TTDD转发

1. 查询转发

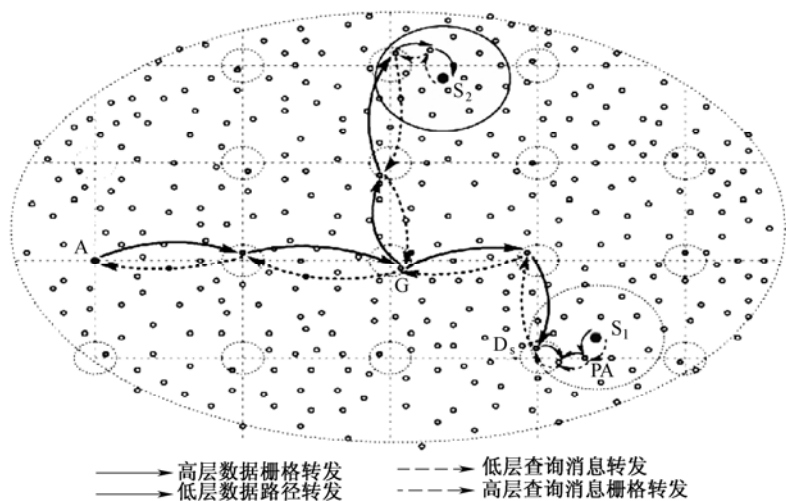
两层查询与数据转发以虚拟栅格基础设施为基础，确保可扩展性和效率。中心节点需要数据时，在其本地区域内泛洪查询消息，寻找附近的分发节点，本地区域约为一个蜂窝般大小。中心节点在查询消息中说明泛洪的最大距离，因此会在离中心节点最大距离左右的节点上停止查询消息的泛洪。

查询消息传递到达一个本地分发节点后（这个分发节点叫做中心节点的直接分发节点），然后被转发到栅格上的上行分发节点，直接分发节点接收这个上行分发节点发送来的数据通知消息。上行分发节点又朝源节点上行转发查询消息，查询消息最终传递到达源节点。在这个转发过程中，每个分发节点存储下行分发节点（接收到这个下行分发节点转发的查询消息）的地址。这个状态随后被用来引导数据朝中心节点传递，如图 6-6（a）所示。

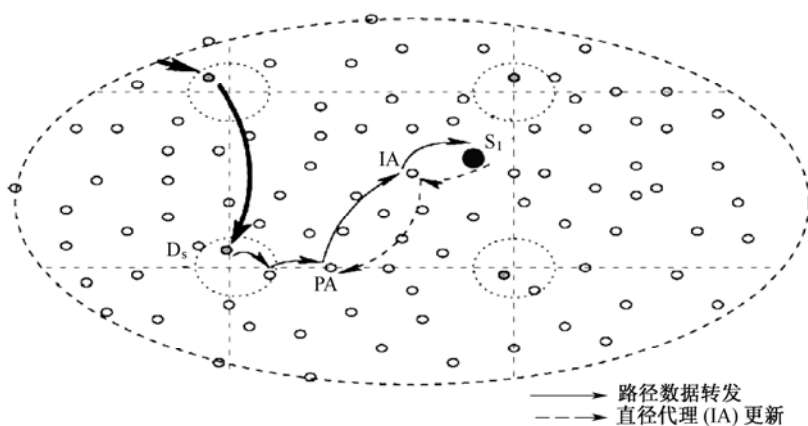
适当采用栅格基础设施能够将查询泛洪限制在单个蜂窝般大小的区域内。相对于在整个传感器场中泛洪查询，这样可以节省大量能量和带宽。而且，在两层转发期间采用两级查询累积可以进一步降低开销。在一个蜂窝内，一个直接分发节点接收到不同中心节点发送的相同数据的查询消息后对其进行累积，然后按照上行更新方式只发送一个备份给其上行分发节点。同理，栅格上的一个分发节点接收到不同下行相邻节点发送来的多个上行更新后，只转发其中一个更新。例如，在图 6-6（a）中，中心节点 S_1 使用其主代理 PA 的位置开始将其查询消息泛洪给其直接分发节点 D_s 。 D_s 记录 PA 的位置信息，然后将查询消息转发给其上行分发节点，直到查询消息传递到达源节点 A 为止。数据沿着查询消息传递路径的反方向回传给 D_s ， D_s 将数据转发给 PA，数据最终到达中心节点 S_1 。中心节点 S_2 的查询消息和数据传输过程与此类似，但是其发送的查询消息将在栅格上的分发节点上被停止传输。因此，分发节点 G 接收到从两个蜂窝（即中心节点 S_1 所在的蜂窝和中心节点 S_2 所在的蜂窝）发送来的查询消息，然后只朝源节点发送一条上行更新消息。

当上行更新消息在栅格中传递的时候，在分发节点中建立软状态，引导数据流回传给中心节点。除非被更新，否则这些软状态只在一定时间周期内有效。分发节点周期性发送上行更新消息，以便连续接收数据；当不再需要数据时，比如当中心节点停止发送查询消息时或者移动离开本地区域的时候，则停止发送上行更新消息。上行分发节点在软状态结束后自动停止转发数据。软状态定时器时间比数据消息间隔时间高一个数量级，这样设置可以平衡周期性产生上行更新消息的开销以及给不再需要数据的区域发送数据的开销。

两级累积适应许多中心节点。查询转发路径上的分发节点只维护有关三个相邻分发节点需要数据的状态信息，而直接分发节点则还需要维护本地单个蜂窝般大小区域内中心节点的状态信息。不参与查询消息和数据转发的传感器节点不保存有关中心节点和数据源节点的任何状态信息。



(a) 在源节点 A 和两个中心节点 S_1 、 S_2 之间的两层查询数据转发



(b) 从直接分发节点 D_s 到移动中心节点 S_1 的路径转发

图 6-6 TTDD 说明图

2. 数据转发

源节点一旦从其某个相邻分发节点接收到查询消息（采用上行更新形式），则立即给这个相邻分发节点发送数据，这个相邻分发节点又将数据转发给自己的一个相邻分发节点（接收到其发送的查询消息），以此一步一步进行，直到数据传递到达中心节点的直接分发节点为止。一个分发节点若是累积了多个不同下行分发节点转发的查询消息，则给每个下行分发节点发送一个数据备份。例如在图 6-6 (b) 中，分发节点 G 给 S_1 和 S_2 发送数据。数据传递到达中心节点的直接分发节点后，采用路径转发将数据转发给中心节点，中心节点可能在连续移动。

采用上述两层转发，可能采取全次佳路径传输查询消息和数据，因此相对于最短路径转发存在额外开销。例如在图 6-6 (b) 中，中心节点 S_1 、 S_2 若是在整个传感器场中泛洪查询消息，则可以沿着直线路径到达源节点。但是，采用两层转发，中心节点和源节点之间的消息

传递路径最多是直线路径的 $\sqrt{2}$ 倍。因此,采用次佳路径获得可扩展性是值得的。

3. 路径转发

采用路径转发将数据从直接分发节点中继给移动中心节点。在路径转发中,每个中心节点跟两个传感器节点有关:主代理和直接代理。中心节点选择一个相邻传感器节点作为其主代理,将主代理的地址信息填写到其查询消息中。直接分发节点将数据发送给主代理,主代理再将数据转发给中心节点。开始时主代理和直接代理是同一个传感器节点。

中心节点准备移动离开当前直接代理覆盖范围时,选择另一个相邻节点作为新的直接代理,将新直接代理的地址发送给主代理,随后的数据被转发给新直接代理。为了避免丢失已经发送给旧直接代理的数据,还要将新直接代理的位置信息发送给旧直接代理[见图 6-6(d)]。中心节点广播请求消息,选择应答信噪比最强的那个相邻节点作为新的直接代理。

主代理代表在中心节点的直接分发节点的移动中心节点,所以中心节点的移动性对其直接分发节点是透明的。直接代理代表在中心节点主代理的中心节点,所以中心节点在持续移动的时候能够连续接收数据。不知道自己位置的中心节点(用户)也仍然能够从 WSN 中收集数据。

中心节点移动离开其主代理一定范围(比如蜂窝大小)后,选择一个新的主代理,本地泛洪查询消息,寻找可能较近的新的分发节点。为了避免接收旧主代理发送的重复数据,TTDD 使每个主代理采用超时机制,超时时间约等于移动中心节点在蜂窝中停留的时间,定时器一旦超时结束则主代理立即消失。旧直接代理采用类似超时机制,不过超时时间较短,约等于中心节点在一跳距离范围内停留的时间。假如中心节点的直接分发节点在一定时间周期(类似于中心节点主代理超时时间)内没有任何其他中心节点或者申请数据的相邻下行分发节点,那么停止给其上行分发节点发送更新消息,不再将数据转发到这个蜂窝中。

一个例子如图 6-6(a)所示。直接分发节点 D_s 的软状态期满结束后, D_s 停止发送上行更新消息,这是因为 D_s 没有任何其他中心节点或者申请数据的相邻下行分发节点。假如中心节点 S_2 需要数据,那么经过一段时间后, G 转发的数据消息只会到达 S_2 。这样,通过在栅格上传递的旧查询消息以及旧代理建立起来的所有状态被清除。

采用路径转发,小范围内(大约一个蜂窝般大小)的中心节点移动性对于高层栅格转发是透明的。大于蜂窝范围的移动性涉及寻找新的分发节点,可能影响栅格上的一些上行分发节点。因为中心节点寻找到的新分发节点很可能在相邻蜂窝内,所以对栅格转发的调整通常只影响附近的几个分发节点。

在图 6-6(b)中,路径转发经过主代理 PA 和直接代理 IA。直接代理 IA 离 S_1 一跳远,将数据直接中继给 S_1 。当 S_1 移动与其当前直接代理 IA 的距离大于一跳后, S_1 从其相邻节点中选择一个新的直接代理 IA,然后给其主代理 PA 和旧 IA 发送一条更新消息,以便中继数据。只要 S_1 移动但仍然处在离 PA 的一定距离范围内,那么 PA 保持不变。

6.2.4 栅格维护

为了避免无限期保持分发节点的状态,源节点在发送数据通知消息、建立栅格时,在数据通知消息中填入栅格寿命。假如栅格寿命结束,那么栅格上的分发节点不会再接收到数据通知消息来更新栅格寿命,清除其状态,栅格不再存在。

适当的栅格寿命值依赖数据有效周期和 WSN 的任务。在图 6-5 中, 假如 WSN 的任务是返回“当前”坦克位置, 那么源节点能够估计坦克周围停留的时间, 并用其来设置栅格寿命。假如坦克停留时间大于原估计时间, 那么源节点可以发送新的数据通知消息来延长栅格寿命。

对于任意结构, 处理意外组件故障、提高强壮性是非常重要的。为了节省传感器的稀少能源, 不在栅格存在期间周期性刷新栅格, 而是采用一种叫做上行信息复制机制——每个分发节点将其上行分发节点的地址信息复制到自己的相邻节点中。当分发节点失效时, 需要数据的下行分发节点发送的上行更新消息将被其中一个相邻节点所接收, 然后这个相邻节点根据所存储的信息又将更新消息转发给上行分发节点。随后上行分发节点发送数据时, 遵循源节点起始构建栅格的方法和规则寻找一个新的分发节点。判定分发节点失效的方法有: 通过 MAC 层机制 (比如 MAC 应答); 假如有一段时间没有旁听到分发节点发送, 则直接要求该分发节点做出应答。

新分发节点不知道相邻的哪个下行分发节点需要数据, 因而将数据转发给所有其他三个分发点。需要数据的下行分发节点将继续发送上行更新消息来重建转发状态; 而不需要数据的下行分发节点将数据丢掉, 不发送上行更新消息, 因而随后的数据不会到达这些分发节点。这种机制还要处理转发路径上多个分发节点同时失效的情况。

中心节点通过超时检测直接分发节点失效问题。中心节点有一段时间没有接收到数据后, 重新泛洪查询消息, 寻找新的分发节点。采用类似超时方法检测主代理和直接代理的失效问题, 选择新的主代理和直接代理。这些技术提高了 TTDD 面对意外节点失效时的强壮性。

通过正在进行的查询或者上行更新按需触发栅格维护。与周期性栅格刷新对比之下, TTDD 对处理开销作了折中, 获取能耗的降低, 能量是 WSN 的较关键资源。

6.2.5 TTDD开销分析

本节分析 TTDD 的效率和可扩展性。测试两个指标: 许多中心节点提取某个源节点一定数量数据的通信开销, 传感器节点为数据分发而维护的状态的复杂性, 分别研究静态中心节点和移动中心节点两种情形。

比较 TTDD 与面向中心节点的数据分发法 (Sink Oriented Data Dissemination, SODD)。在 SODD 中, 每个中心节点首先向全网泛洪, 在所有传感器节点上建立数据转发状态, 然后源节点响应中心节点而向其交付数据。定向扩散采用 SODD 法, 但是采用了不同的优化技术, 比如数据累积和查询累积, 减少所交付的消息数量。因为数据累积和查询累积技术也适用于 TTDD, 所以在进行通信开销分析时没有考虑数据累积和查询累积。重点在于研究和分析每种协议在最差情形下的通信开销。目的是为了保持分析简单, 易于理解, 同时又能捕捉到 TTDD 和 SODD 之间的基本不同点。在分析传感器状态维护复杂性时考虑累积的影响。

1. 模型与符号

考虑一个正方形传感器场区域 A , 其中均匀分布 N 个传感器节点, 因此每边大约 \sqrt{N} 个传感器节点。场中存在 k 个中心节点, 以平均速度 v 移动, 同时在时间周期 T 内从一个源节点接收 d 个数据分组。每个数据分组大小为一个单位, 查询消息、数据通知消息的长度均为 l 。在一个区域泛洪的通信开销与其中传感器节点数量成正比。沿着一条路径采用贪婪地理路

由转发消息的通信开销与该路径上的传感器节点数量成正比。一个传感器节点无线通信范围内的平均相邻节点数为 D 。

在 TTDD 中, 源节点将传感器场分成一个一个的蜂窝: 每个蜂窝是大小为 α^2 的区域 (正方形), 存在 $n=N\alpha^2/A$ 个传感器节点, 蜂窝的每边存在 \sqrt{n} 个传感器节点。每个中心节点通过 m 个蜂窝, m 取 $1+vT/\alpha$ 的上限。对于静态中心节点, $m=1$ 。

2. 通信开销

首先分析 TTDD 和 SODD 在最差情形下的通信开销。假定在 TTDD 和 SODD 中, 中心节点在连续两次位置更新之间更新其位置 m 次, 接收 d/m 个数据分组。在 TTDD 中, 中心节点通过本地泛洪查询消息来更新其位置, 使查询消息到达直接分发节点, 直接分发节点沿着栅格进一步将查询消息转发给源节点。在不考虑查询累积下, 查询消息传递到达源节点的开销为 $nl + \sqrt{2}(c\sqrt{N})l$, 其中 nl 表示本地泛洪开销, $c\sqrt{N}$ 表示从源节点沿着一条直线路径到达中心节点所经过的平均传感器节点数量 ($0 < c \leq \sqrt{2}$)。因为在 TTDD 中查询消息传递经过栅格而不是经过一条直线路径, 所以最差情形路径长度增大到 $\sqrt{2}$ 倍。

类似地, 源节点给中心节点交付 d/m 个数据分组的通信开销为 $\sqrt{2}(c\sqrt{N})d/m$ 。对于 k 个移动中心节点, 在 k 个蜂窝中接收 d 个分组的通信开销为

$$km \left[nl + \sqrt{2}(c\sqrt{N})l + \sqrt{2}(c\sqrt{N})d/m \right] = kmnl + kc(ml + d)\sqrt{2N}$$

加上 WSN 任务更新的通信开销 Nl 、构建栅格的通信开销 $\frac{4N}{\sqrt{n}}l$, TTDD 的总通信开销 (CO) 为

$$CO_{TTDD} = Nl + \frac{4N}{\sqrt{n}}l + kmnl + kc(ml + d)\sqrt{2N} \quad (6-19)$$

在 SODD 中, 每次中心节点对整个网络泛洪时, 中心节点接收 d/m 个数据分组。数据分组沿着直线路径传递到达中心节点。不考虑数据累积, 通信开销为 $Nl + (c\sqrt{N})d/m$ 。对于 k 个移动中心节点, 最差情形的总通信开销为 $CO_{SODD} = km \left[Nl + (c\sqrt{N})d/m \right] = kmNl + kcd\sqrt{N}$ 。

注意这里没有考虑 WSN 任务更新的开销, 这是因为 SODD 在泛洪查询消息时能够潜在地更新 WSN 的任务。

比较 TTDD 和 SODD 的通信开销, 得到 $\frac{CO_{TTDD}}{CO_{SODD}} \approx \frac{1}{mk} \left(1 + \frac{4}{\sqrt{n}} \right), N \dots n, \left(\frac{d}{m} \right)^2$ 。

因此, 在大规模 WSN 中, 随着 WSN 规模 (N) 的增大、中心节点的增多 (k) 以及中心节点移动性 (由 m 来描述) 的增强, TTDD 的最差情形通信开销渐近低于 SODD。

例如, 一个 WSN 由 $N=10\,000$ 个传感器节点组成, 在一个 TTDD 栅格蜂窝中存在 $n=100$ 个传感器节点。设 $c=1$ 、 $l=1$, 交付 $d=100$ 个数据分组, 则 $\frac{CO_{TTDD}}{CO_{SODD}} = \frac{0.024m + 1.4 \cdot \frac{1}{k} + 1.414}{m + 1}$ 。

对于静态中心节点, $m=1$, 假定有四个中心节点 $k=4$, 则 $CO_{TTDD}/CO_{SODD}=0.89$ 。当中心节点移动性增强的时候, 随着 $m \rightarrow \infty$, $CO_{TTDD}/CO_{SODD} \rightarrow 0.024$ 。在这种网络设置下, TTDD 在

静态中心节点和移动中心节点下的通信开销总是低于 SODD。

式 (6-19) 表明一个蜂窝中的传感器节点数量 (n) 对 TTDD 通信开销的影响。对于上述例子, 图 6-7 (d) 给出了 TTDD 在不同中心节点移动速度下的通信开销与 n 之间的变化关系曲线。由于随着蜂窝的增大, 构建栅格的通信开销随着下降, 而本地查询泛洪开销却随着增大, 所以图 6-7 (d) 说明总的通信开销是栅格构建开销和本地查询泛洪开销之间的折中平衡。从图 6-7 (d) 中还可以看到: 当中心节点移动性非常重要时, 蜂窝越小, 则总通信开销越低, 其原因是强移动性导致频繁的蜂窝内泛洪, 蜂窝越小则泛洪开销越低。

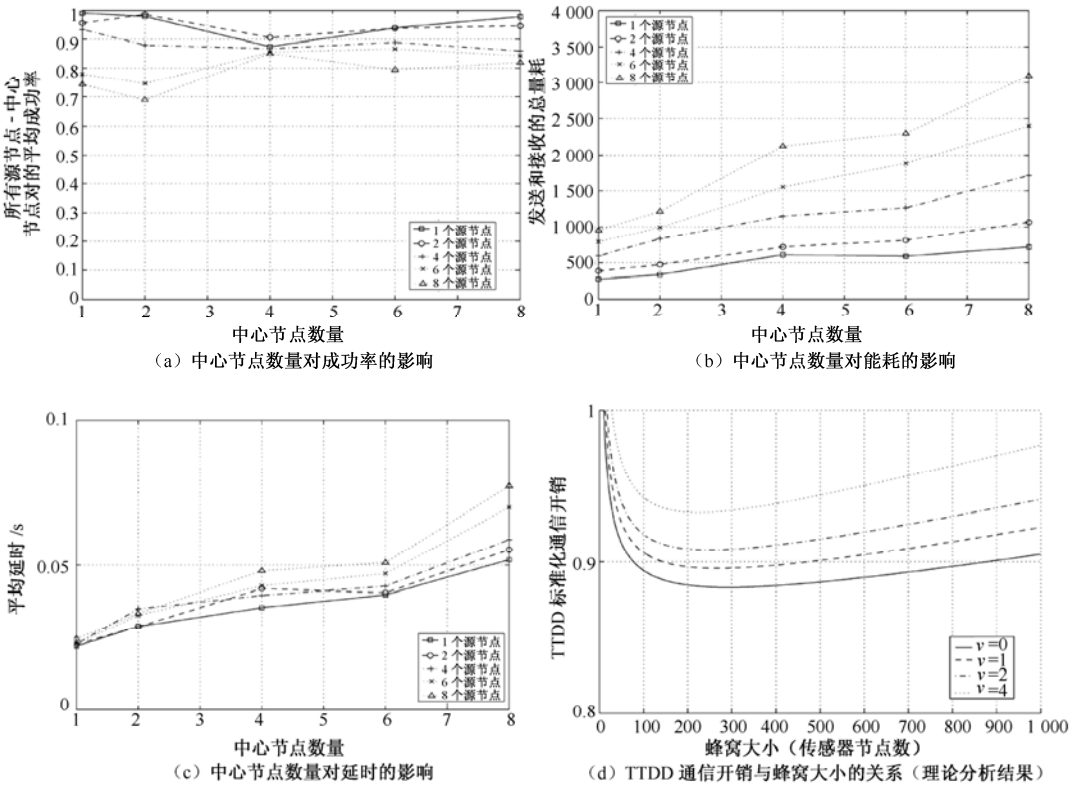


图 6-7 TTDD 的性能

3. 状态复杂性

在 TTDD 中, 只有分发节点及其复制上行信息的相邻节点、中心节点的主代理和直接代理才维护数据分发的状态。所有其他传感器节点不需要维护任何状态。不同传感器节点的状态复杂性如下所述:

- ① 分发节点: 一个栅格上总共存在 $(\sqrt{N/n}+1)^2$ 个分发节点, 每个分发节点为查询消息转发而维护其上行分发节点的位置信息。对于数据转发路径上的分发节点, 每个分发节点为数据转发而最多维护另外三个相邻分发节点的位置信息。因此, 一个分发节点的状态复杂性为 $O(1)$ 。复制上行分发节点的位置信息的分发节点的相邻节点的状态复杂性也为 $O(1)$ 。
- ② 直接分发节点: 一个直接分发节点维护本地蜂窝般大小区域内有关所有中心节点的主代理的状态。假定本地区域内存在 k_{local} 个中心节点, 因此一个直接分发节点的状态复杂性

为 $O(k_{\text{local}})$ 。

③ 主代理和直接代理：主代理维护其中心节点的直接代理的位置信息，直接代理维护其中心节点的路径转发信息。两者的状态复杂性均为 $O(1)$ 。

④ 源节点：源节点维护其栅格大小信息、申请数据的下行分发节点的位置信息，因此其状态复杂性为 $O(1)$ 。

考虑从 s 个源节点将数据转发到 k 个移动中心节点。假定在 SODD 中，从一个源节点到所有中心节点的数据转发路径上的传感器节点总数为 P ；那么在 TTDD 栅格转发路径上的传感器节点数量最多为 $\sqrt{2}P$ 。中心节点直接分发节点、主代理、直接代理为路径转发而维护的状态总量为 $k(s+2)$ 。总状态复杂性为 $s \left[b \left(\sqrt{\frac{N}{n}} + 1 \right)^2 + 3\sqrt{2} \frac{P}{\sqrt{n}} \right] + k(s+2)$ ，其中 b 表示一个分发点周围的传感器节点数量，这个分发点具有上行分发节点的位置信息， b 是一个取值较小的常数。

在 SODD 中，每个传感器节点维护从其相邻节点到达源节点的状态。对于多个源节点，假定进行完全的数据累积，那么一个传感器节点最多维护每个相邻节点的状态。对于转发路径上的传感器节点，由于查询累积，所以最多维护每个相邻节点的状态，在多个中心节点情况下引导数据的传递。

整个 WSN 的状态复杂性为 $(D-1)N + (D-1)P = (D-1)(N+P)$ 。
TTDD 与 SODD 的状态复杂性之比为 $S_{\text{TTDD}}/S_{\text{SODD}} \rightarrow sb/n(D-1)$ ， $N \rightarrow \infty$ 。

因此，对于大规模 WSN，TTDD 维护的状态只占 SODD 维护状态的 $sb/n(D-1)$ 。对于图 6-5，2 个源节点，3 个中心节点，假设 $b=5$ ，一个 TTDD 栅格蜂窝内存在 100 个传感器节点，每个传感器节点平均 10 个相邻节点，那么 TTDD 维护的状态是 SODD 的 1.1%。

4. 分析小结

上面分析了 TTDD 在最差情形下的通信开销和状态复杂性。与 SODD 法对比，TTDD 在最差情形下的通信开销随着 WSN 规模的扩大、中心节点的增多、中心节点移动速度的增大而渐近低于 SODD。TTDD 的状态复杂性低于 SODD，这是因为不在栅格基础设施上的传感器节点不需要维护数据分发的状态。对于栅格基础设施上的传感器节点，其状态复杂性受到限制，但是与 WSN 规模、源节点和中心节点的数量无关。

6.2.6 TTDD 的性能

下面介绍通过仿真来评估 TTDD 的性能。仿真结果证实了 TTDD 将多个源节点的数据交付给多个移动中心节点的效率和可扩展性，TTDD 在静态中心节点下的性能与定向扩散相当。

1. 性能指标与评估方法

在 ns-2 中实现 TTDD。采用基本的贪婪地理转发及本地泛洪来绕过失效节点。为了比较 TTDD 与定向扩散，采用 ns-2.1b8a 中 TTDD 实现时的相同能量模型。采用 IEEE 802.11 DCF 作为低层 MAC 协议。一个传感器节点的发射功耗为 0.66 W，接收功耗为 0.395 W，空

闲功耗为 0.035 W。

采用三个性能指标参数评估 TTDD。① 能耗：定义为 WSN 的通信（发送和接收）能耗。空闲能耗主要依赖数据产生间隔时间，并不表示数据交付效率，所以计算能耗时不包括空闲能耗。② 成功率：表示一个中心节点成功接收的数据分组数量与一个源节点产生的数据分组总数量之比，求所有源节点-中心节点对的平均成功率。成功率表示数据交付的效率。③ 时延：定义为一个源节点数据分组发送时刻与一个中心节点该数据分组接收时刻之间的平均时间，求所有源节点-中心节点对的平均时延。时延表示数据分组的新鲜程度。

默认仿真设置是：4 个中心节点，200 个传感器节点随机分布在 $2\,000\times2\,000\text{ m}^2$ 的场中，其中 4 个源节点，每次仿真持续时间 200 s，每个结果是 6 个随机网络拓扑实验结果的平均值，所有随机拓扑由 ns-2 中的 setdest 工具生成，每个源节点每秒产生一个分组，中心节点的移动遵循标准随机点移动模型，每个查询分组 36 B，每个数据分组 64 B，蜂窝大小参数 α 设为 600 m，中心节点的本地查询泛洪范围设为 1.3α 以便于处理不规则的分发节点分布。

2. 中心节点和源节点数量的影响

中心节点和源节点的数量变化范围为 1、2、4、6、7、8 个，中心节点最大移动速度为 10 m/s，暂停时间为 5 s。

图 6-7（a）给出了成功率实验结果。对于图 6-7（a）中每条不同源节点数量下的成功率曲线，成功率随着中心节点数量的变化而波动。但是几乎所有的成功率都处在 0.8~1.0 的范围内。对于特定数量的中心节点，成功率随着源节点的增多而下降。在 8 个中心节点条件下，成功率随着源节点增加到 8 个而从 1.0 下降到约 0.8，这是因为源节点越多，产生的数据分组越多，从而导致竞争引起的分组丢失越多。总之，成功率实验结果表明：TTDD 将多个源节点的大部分数据成功交付给多个移动中心节点，且交付质量没有随着源节点或者中心节点的增多而下降过多。

图 6-7（b）给出了能耗实验结果。对于图 6-7（b）中每条不同源节点数量下的能耗曲线，能耗随着中心节点的增多而缓慢、次线性地增大，这是因为中心节点越多，本地泛洪查询消息越多，涉及数据转发的分发节点越多，两者的能耗就越多。但是能耗与中心节点数量呈次线性递增关系的原因是多个中心节点针对同一个源节点发送的查询消息在高层栅格转发时被组合在一起。对于特定数量的中心节点（比如 4 个中心节点），能耗几乎随着源节点的增多而线性递增，这是因为源节点产生的总数据分组数量按比例递增，导致能耗成比例增大。一个例外是：当源节点从一个增加到两个时，能耗递增较少，这是因为在一个源节点的时候，低层查询泛洪的能耗占总能耗中的一大部分，但是随着源节点的增多，低层查询泛洪的能耗却保持不变。

图 6-7（c）给出了时延实验结果，其范围在 0.02~0.08 s 之间。中心节点越多，源节点越多，时延越大。源节点越多，产生的数据分组越多；中心节点越多，本地泛洪的查询消息越多，这两者导致流量增大以及数据分组交付时延增大。而且，即使在 8 个中心节点和 8 个源节点条件下，时延也相对很小。

3. 中心节点移动性的影响

接着评估中心节点移动性对 TTDD 性能的影响。在默认设置中，改变中心节点的最大移

动速度, 其变化范围为 0 m/s, 5 m/s, 10 m/s, 15 m/s, 20 m/s。仿真结果如下:

① 中心节点较快速移动的时候, 成功率保持在 85% 左右。这就表明中心节点能够迅速适应其位置的变化, 即使在高达 20 m/s 的移动速度下, 中心节点也能够从新的代理和 (或者) 新的分发节点接收数据分组。

② 能耗随着中心节点移动速度的增大而增大, 中心节点移动速度越高, 中心节点本地泛洪查询消息、寻找新的直接分发节点就越频繁。但是, 曲线的斜率随着中心节点移动速度的增大而下降, 这是因为高层栅格转发只随着中心节点移动而递增式变化。

③ 数据交付时延随着中心节点移动速度的增大, 从 0.03 s 稍微增大到 0.045 s。这个结果说明: 高层栅格转发能够有效地使中心节点移动性的影响本地化。

4. 抗传感器节点失效能力

在默认设置的 200 个传感器节点中, 最多随机选择 15% 的传感器节点在时刻 $t=20$ s 时同时失效。详细的仿真跟踪研究表明: 在这种实验方案下, 栅格上有些分发节点失效。不作任何恢复, 分发节点失效将停止将数据交付给所有下行中心节点, 导致成功率大幅下降。但是, 实验结果表明成功率适度下降。这就证明 TTDD 栅格维护机制对降低节点失效带来的损害是有效的。当节点失效变得越来越严重的时候, 由于交付数据的减少, 数据交付能耗随着下降; 另一方面, 中心节点寻找备用分发节点的能耗却随着增大, 两者的综合效果就是能耗稍有下降。由于修复失效分发节点需要时间, 所以随着越来越多节点失效, 平均时延稍有递增。总之, TTDD 在所有仿真实验中表现出很强的抗节点失效能力。

5. 蜂窝大小 α 的影响

前面已经介绍了各种环境因素对 TTDD 性能的影响, 现在介绍控制参数 (蜂窝大小 α) 对 TTDD 性能的影响。为了将蜂窝扩大而传感器场中又仍然有足够多的蜂窝, 假如保持节点密度不变, 那么就必须仿真 2 000 个以上的传感器节点。给定 ns-2 的可用计算能力, 必须降低节点密度, 以便减少仿真传感器节点总数。在 $6\,200 \times 6\,200\text{ m}^2$ 场中采用 961 个传感器节点, 传感器节点有规则、200 m 等距离分布, 以便仍然能够采用简单的贪婪地理转发功能。一个源节点, 一个中心节点。中心节点以 10 m/s 恒定速度移动。蜂窝大小变化范围 400~1 600 m, 递增步长 200 m。由于传感器节点规则布置, 所以成功率和时延变化不大。因此, 重点考虑能耗。

能耗首先随着蜂窝增大而下降, 这是因为构建较大蜂窝栅格的能耗较低; 但是蜂窝一旦增大到 1 000 m, 那么能耗开始增大, 这是因为在大蜂窝中本地泛洪查询消息的能耗较高。假如整个传感器场只有一个蜂窝, 则本地查询泛洪变成全网查询泛洪。

6. 与定向扩散协议的对比

对 TTDD、定向扩散采用相同拓扑进行实验, 中心节点保持静止不动。按照前面那样改变中心节点和源节点的数量, 研究 TTDD、定向扩散对中心节点和源节点的可扩展性。所有仿真实验采用随机分布在 $2\,000 \times 2\,000\text{ m}^2$ 区域中的 200 个传感器节点。仿真结果见图 6-8 (a) ~ 图 6-8 (f)。

图 6-8 (a) 和图 6-8 (d) 分别给出 TTDD、定向扩散的成功率实验结果。TTDD 和定向

扩散具有类似的成功率，范围在 0.7~1.0 之间。静态中心节点下的 TTDD 成功率不如移动中心节点下的 TTDD 成功率，这是因为没有源节点分发节点的静态中心节点不能移动到其他地点来寻找分发节点。在有些情况下，移动性也有助于数据分发。

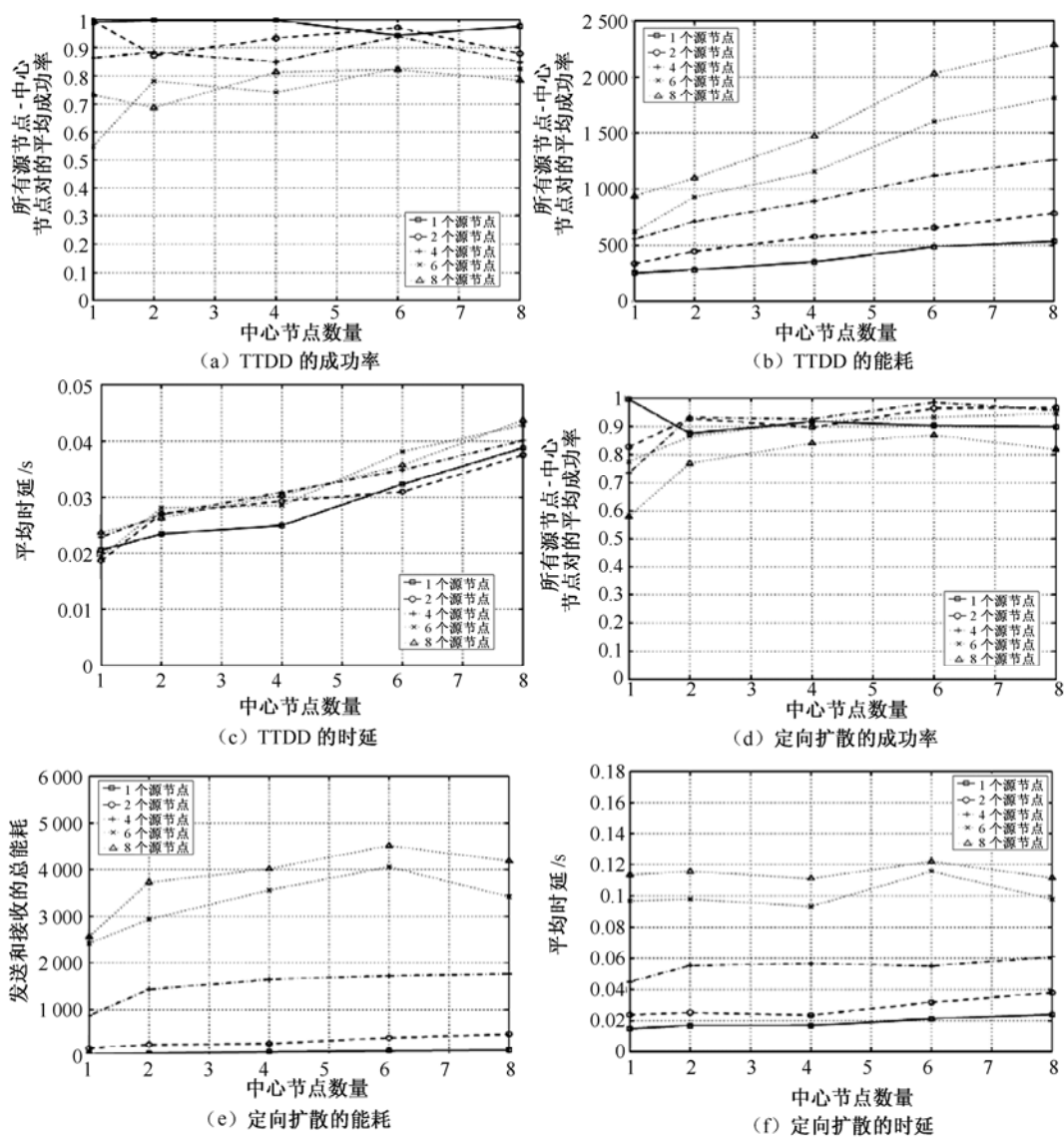


图 6-8 TTDD、定向扩散在静态中心节点下的性能对比

图 6-8 (b) 和图 6-8 (e) 分别给出 TTDD、定向扩散的能耗实验结果。在 1~2 个源节点时，定向扩散的能耗低于 TTDD；当源节点多于 2 个时，TTDD 的能耗比定向扩散低得多。这就说明 TTDD 的源节点可扩展性优于定向扩散。在定向扩散中，没有针对特定源节点的专门节点集，所有源节点共享所有传感器节点，将数据交付给中心节点。但是，TTDD 采用专门技术，将总的的分发载荷分散到两层上。每个源节点构建自己的栅格，专门用于自己的数据分发。不同的源节点使用不同的栅格，相互间的干扰达到最小。对于相同数量的源节点，定向扩散较主动累积不同中心节点的查询消息，因此当中心节点较多时，能耗不会迅速增大。

在图 6-8 (e) 中, 对于定向扩散, 当中心节点从 6 个增加到 8 个时, 能耗下降异常, 这是因为当检测到低交付质量时, 定向扩散源节点停止产生数据分组。在这两种情形下, 产生较少的数据流量, 因此总能耗下降。

图 6-8 (c) 和图 6-8 (f) 分别给出 TTDD、定向扩散的时延实验结果。在 1~2 个源节点时, TTDD、定向扩散的时延相当。继续增加源节点, TTDD 的时延增大速度比定向扩散低得多。这是因为在定向扩散中, 不同源节点的数据转发路径可能相互重叠、相互干扰, 特别是源节点比较多时这种情况更加严重; 而在 TTDD 中, 每个源节点有自己的栅格, 数据沿着不同栅格传递, 相互干扰没有定向扩散那样重。

6.2.7 TTDD讨论

(1) 蜂窝大小信息

传感器节点需要知道蜂窝大小 α , 这样一旦自己成为源节点就能够构建栅格。可以采用某种外部机制说明 α 。一种实现方法是使任务陈述消息包含参数 α , 该消息通知每个传感器节点执行的感知任务。在网络开始操作的时候或者在任务更新期间将任务陈述消息泛洪到每个传感器节点。中心节点也需要参数 α 来指定查询消息被泛洪的范围。中心节点可以从其相邻节点获取参数 α 。为了处理不规则的本地拓扑, 分发节点可能位于固定泛洪区域之外, 中心节点可以采用扩展环形搜索法搜索附近的分发节点。

(2) 贪婪地理路由中断

贪婪地理转发路由可能中断, 即不存在需要临时转发分组的贪婪路径。可以采用一种简单技术强化贪婪转发路径, 当贪婪路径中断不存在时, 即分组被转发到一个没有相邻节点且比较接近目的节点的传感器节点, 该传感器节点本地泛洪该分组, 以便绕过这个死胡同。

此外, 由于传感器节点随机分散, 所以在有些情况下, 采用地理贪婪转发, 节点 A 的分组能够成功传递到达节点 B, 但是节点 B 的分组却不能传递到达节点 A。这种不对称转发导致有些分发节点的上行更新分组被丢失而无法传递给自己的上行分发节点, 因此没有下行中心节点的数据。前面提到的超时技术能够减轻这个问题, 帮助中心节点寻找其他能够成功发送上行更新的直接分发节点。一般地, 当成功率非常关键时, 应该采用贪婪路由中断问题的完整解决方法。

(3) 移动因素

TTDD 重点是处理中心节点的移动性。在移动因素激励下, 源节点沿着移动轨迹可以各自构建一个栅格。为了避免频繁构建栅格, 源节点可以重复利用其他源节点已经建立的栅格。采用中心节点寻找直接分发节点的方法寻找已建立的栅格。特别是当源节点有数据需要发送时, 源节点在蜂窝般大小的区域内泛洪“栅格寻找”消息, 检测现有的栅格。现有栅格上的分发节点应答新的源节点。然后源节点就可以使用现有栅格分发自己的数据。

(4) 非均匀栅格布置

迄今为止没有假定事先知道中心节点的位置。因此, 构建均匀栅格, 尽可能均匀分发转发状态。但是, 均匀分布的一个缺点是: 对于中心节点不会漫游进入的区域, 存在一定的资源浪费。通过获知或者预测中心节点的位置, 可以部分解决这个资源浪费问题。假如可以使用中心节点的位置, 那么可以进一步优化 TTDD, 构建一个完全非均匀栅格, 只有中心节点当前驻留区域或者移动区域才会有栅格。中心节点当前位置以及随后位置的估计

精度影响性能。

(5) 中心节点的移动速度

TTDD 将中心节点移动性对数据分发的影响限制在一个蜂窝范围内，通过路径转发处理蜂窝内的移动性。但是，TTDD 容纳中心节点移动性也是有限制的。中心节点的移动速度不能快于本地转发状态的更新速度（在一个蜂窝区域内）。两层转发很适合处理“本地化”移动模式，在该模式中，中心节点不会频繁改变其主代理。

(6) 栅格自维护

采用上行信息复制机制处理分发节点意外失效问题。在每个分发节点周围的一跳相邻节点中复制栅格状态。当极少发生分发节点失效问题时，为进一步排除状态维护的冗余度，可以重复使用递归栅格构建机制，以便栅格能够自行维护。特别是，当进入一个所有分发节点全部失效的“空”区域构建栅格时，宜用查询消息或者数据分组。这样，正在泛洪中的查询消息或者正在转发中的数据分组对栅格结构修复起到数据通知消息的作用。

参 考 文 献

- [1] A. Chandrakasan, R. Amirtharajah, S.-H. Cho, J. Goodman, G. Konduri, J. Kulik, W. Rabiner, and A. Wang. Design considerations for distributed microsensor systems. in Proc. IEEE Custom Integrated Circuits Conf. (CICC), San Diego, CA, May 1999, pp.279–286.
- [2] L. Clare, G. Pottie, and J. Agre. Self-organizing distributed sensor networks. in Proc. SPIE Conf. Unattended Ground Sensor Technologies and Applications, vol. 3713, Orlando, FL, Apr. 1999, pp.229–237.
- [3] Wendi B. Heinzelman, Anantha P. Chandrakasan, and Hari Balakrishnan. An Application-Specific Protocol Architecture for Wireless Microsensor Networks. in IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, Vol.1, No.4, pp.660-670, October 2002.
- [4] W. Heinzelman. Application-specific protocol architectures for wireless networks. Ph.D. dissertation, Mass. Inst. Technol., Cambridge, 2000.
- [5] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient routing protocols for wireless microsensor networks. in Proc. 33rd Hawaii Int. Conf. System Sciences (HICSS), Maui, HI, Jan. 2000.
- [6] T. Murata and H. Ishibuchi. Performance evaluation of genetic algorithms for flowshop scheduling problems. Proc. 1st IEEE Conf. Evolutionary Computation, vol.2, pp.812–817, June 1994.
- [7] T. Rappaport, Wireless Communications: Principles & Practice. Englewood Cliffs, NJ: Prentice-Hall, 1996.
- [8] A. Wang, W. Heinzelman, and A. Chandrakasan. Energy-scalable protocols for battery-operated microsensor networks. Proc. 1999 IEEE Workshop Signal Processing Systems (SiPS '99), pp.483–492, Oct.1999.
- [9] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton and J. Zhao. Habitat monitoring: Application driver for wireless communications technology. Proceedings of ACM SIGCOMM Workshop on Data Communications in Latin America and Caribbean (2001).

- [10] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler and K. Pister. System architecture directions for networked sensors. Proceedings of International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-IX) (2000).
- [11] FAN YE, HAIYUN LUO, JERRY CHENG, SONGWU LU and LIXIA ZHANG. A Two-Tier Data Dissemination Model for Large-scale Wireless Sensor Networks. Proceedings of ACM International Conference on Mobile Computing and Networking (MOBICOM'02) (2002).
- [12] HAIYUN LUO, FAN YE, JERRY CHENG, SONGWU LU and LIXIA ZHANG. TTDD: Two-Tier Data Dissemination in Large-Scale Wireless Sensor Networks. *Wireless Networks* 11, 161-175, 2005.
- [13] B. Karp and H. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. Proceedings of ACM International Conference on Mobile Computing and Networking (MOBICOM'00) (2000).
- [14] G. Pottie and W. Kaiser, Wireless integrated network sensors, *Communications of the ACM* 43(5) (2000) 51–58.
- [15] F. Ye, S. Lu and L. Zhang. GRAdient Broadcast: A robust, long-lived large sensor network (2001), <http://irl.cs.ucla.edu/papers/grab-tech-report.ps>
- [16] Y. Yu, R. Govindan and D. Estrin. Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks. Technical Report UCLA/CSD-TR-01-0023, UCLA Computer Science Department (May 2001).
- [17] V. Kawadia and P.R. Kumar. Power Control and Clustering in Ad Hoc Networks. *Proc. IEEE INFOCOM*, Apr. 2003.
- [18] S. Narayanaswamy, V. Kawadia, R.S. Sreenivas, and P.R. Kumar. Power Control in Ad-Hoc Networks: Theory, Architecture, Algorithm and Implementation of the COMPOW protocol. *Proc. European Wireless 2002. Next Generation Wireless Networks: Technologies, Protocols, Services and Applications*, pp.156-162, Feb.2002.
- [19] Seema Bandyopadhyay and Edward J. Coyle. An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks. *IEEE INFOCOM 2003*, pp.1713-1723, 2003.
- [20] Ossama Younis and Sonia Fahmy. HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks. *IEEE TRANSACTIONS ON MOBILE COMPUTING*, Vol.3, No.4, pp.366-379, 2004.
- [21] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. An Application-Specific Protocol Architecture for Wireless Microsensor Networks. *IEEE Trans. Wireless Comm.*, vol.1, no.4, pp.660-670, Oct. 2002.
- [22] C.R. Lin and M. Gerla. Adaptive Clustering for Mobile Wireless Networks. *IEEE J. Selected Areas Comm.*, Sept. 1997.
- [23] S. Banerjee and S. Khuller. A Clustering Scheme for Hierarchical Control in Multi-Hop Wireless Networks. *Proc. IEEE INFOCOM*, Apr.2001.
- [24] S. Basagni. Distributed Clustering Algorithm for Ad-Hoc Networks. *Proc. Int'l Symp. Parallel Architectures, Algorithms, and Networks (I-SPAN)*, 1999.
- [25] M. Chatterjee, S.K. Das, and D. Turgut. WCA: A Weighted Clustering Algorithm for Mobile

- Ad Hoc Networks. Cluster Computing, pp.193-204, 2002.
- [26] S. Bandyopadhyay and E. Coyle. An Energy-Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks. Proc. IEEE INFOCOM, Apr.2003.
 - [27] A.D. Amis, R. Prakash, T.H.P. Vuong, and D.T. Huynh. Max-Min D-Cluster Formation in Wireless Ad Hoc Networks. Proc. IEEE INFOCOM, Mar. 2000.
 - [28] O. Younis and S. Fahmy. Distributed Clustering in Ad-Hoc Sensor Networks: A Hybrid, Energy-Efficient Approach. Proc. IEEE INFOCOM, Mar.2004.
 - [29] F. Ye, G. Zhong, S. Lu, and L. Zhang. PEAS: A Robust Energy Conserving Protocol for Long-Lived Sensor Networks. Proc. Int'l Conf. Distributed Computing Systems (ICDCS), 2003.
 - [30] D.M. Blough and P. Santi. Investigating Upper Bounds on Network Lifetime Extension for Cell-Based Energy Conservation Techniques in Stationary Ad Hoc Networks. Proc. ACM/IEEE Int'l Conf. Mobile Computing and Networking (MOBICOM), 2002.
 - [31] D.B. Johnson and D.A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. Mobile Computing, vol. 353, 1996, citeseer.ist.psu.edu/johnson96dynamic.html.
 - [32] P. Gupta and P.R. Kumar. Critical Power for Asymptotic Connectivity in Wireless Networks. Stochastic Analysis, Control, Optimizations, and Applications: A Volume in Honor of W.H. Fleming, W.M. McEneaney, G. Yin, and Q. Zhang (Eds.), 1998.
 - [33] S . Shakkottai, R. Srikant, and N. Shroff. Unreliable Sensor Grids: Coverage, Connectivity and Diameter. Proc. IEEE INFOCOM, Mar.2003, citeseer.ist.psu.edu/shakkottai03unreliable. html.
 - [34] S. Shakkottai, R. Srikant, and N. Shroff. An Energy-Efficient Clustering Algorithm for Multihop Data Gathering in Wireless Sensor Networks. JOURNAL OF COMPUTERS, Vol.1, No.1, pp.40-47, April 2006.

第 7 章 无线传感器网络地理位置路由协议

7.1 定位技术

对于军事应用或者营救应用的 Ad Hoc 网络以及 WSN，定位都是不可缺少的。有了可用位置信息，不仅可以提高协议效率，而且可以开发许多新的应用。

节点位置或者物理现象位置就是感知应用的目标。比如在诸如气象监视、环境监视、包裹跟踪、图书馆档案管理之类的应用中，位置是主要内容之一。在 WSN 中，有了节点的位置，才能使每个感知数据关联特定地理区域，节点报告的感知数据才有意义。位置感知和方向感知能够跟踪通过传感器场的移动目标。

全球定位系统（Global Positioning System, GPS）可以提供位置信息。在 GPS 中，使用接收机来估计移动 Ad Hoc 网络节点的位置。但是，GPS 成本高，并且许多 WSN 要求的位置估计精度较高，这就使得 GPS 不适用于 WSN。GPS 使用原子钟进行时间同步。每个 GPS 卫星给地面传感器节点发送该节点位置和当前时间信号。地面传感器节点估计卫星信号传输到达自己所花费的时间，并据此估计自己到达每个 GPS 卫星的距离。一个地面传感器节点一旦完成到达四个 GPS 卫星的距离估计，则可以计算出自己的三维位置。

当前大多数传感器位置检测技术包括两个阶段：① 测量阶段——测量传感器与中心节点之间的距离或者角度；② 计算阶段——将距离或者角度测量结果转换成位置估计。有些位置检测计算在得到初始位置估计后还有一个改进阶段。

7.1.1 距离测量与角度测量

距离估计技术包括到达时间（Time-of-Arrival, ToA）、到达时间差（Time-Difference-of-Arrival, TDoA）、接收信号强度指示器（Received-Signal-Strength-Indicator, RSSI）、到达角度（Angle-of-Arrival, AoA）。AoA 测量信号到达中心节点和传感器的角度。假如中心节点和传感器之间不能直接通信，则可以利用网络连通性进行距离估计。

ToA 和 TDoA 分别测量信号到达中心节点和传感器的时间、时间差，然后根据传输时间和传输速率计算距离。ToA、TDoA 适用于许多不同类型的信号，比如 RF 信号、声学信号、超声波信号等。对比 TDoA，ToA 存在处理时延、非视距传播会产生误差等缺点。ToA 要求时间同步才能精确测量空中传播时间。

RSSI 根据发射功率、接收功率、无线传播模型计算距离，主要用于 RF 信号。由于室外环境中的多径衰落，所以采用 RSSI 进行距离估计可能不精确。

AoA 由于其后续计算（三角测量法）简单而很有吸引力。假如一个传感器被散射目标所包围，那么可能很难精确测量这个传感器。测量 AoA 要求传感器和中心节点配备定向天线或者天线阵，而由于成本和形状原因这可能是不允许的。

假如传感器不能接收到足够多中心节点的信号（对于 AoA 至少需要 2 个中心节点，对于 ToA、TDoA、RSSI 至少需要 3 个中心节点），那么 AoA、ToA、TDoA、RSSI 就不能正常工作，此时可以利用网络连通性进行距离估计。

7.1.2 位置计算

三角测量法、三边测量法、多边测量法三种技术联合利用距离测量和角度测量计算位置，其中三角测量法最简单。如图 7-1（a）所示，假如已知到达中心节点 A 和 B 的角度（ α 和 β ），那么 S 的位置就是直线 AS、BS 的交叉点。因此，对于 AoA，至少需要 2 个中心节点。三边测量法计算 3 个圆的交叉点，如图 7-1（b）所示。假如到达每个中心节点的距离不准确，那么 3 个圆可能没有公共交叉点，因而三边测量法得不到确切答案。多边测量法运用目标函数使传感器的估计位置与真实位置之差最小。例如，在图 7-1（c）中，可以运用函数 $\min \sum_i (D_{S_i} - \bar{D})^2$ 计算 S 的位置 (x,y) ，其中 $D_{S_i} = \sqrt{(x-x_i)^2 + (y-y_i)^2}$ ， \bar{D} 是从 S 到达 i 的距离估计，i 分别为 A、B、C、D、E。多边测量法能够提高位置精度，但是计算开销较高。三边测量法和多边测量法至少需要 3 个中心节点。

基于时间的定位方案（Time-based Positioning Scheme，TPS）根据 TDoA 计算距离，没有同步要求，实质上是检测一个传感器到达 3 个中心节点（其中一个称为主中心节点）的距离差。TPS 采用三边测量法和距离差信息，同时计算传感器的位置及其到达中心节点的距离。下面将详细介绍 TPS 技术。

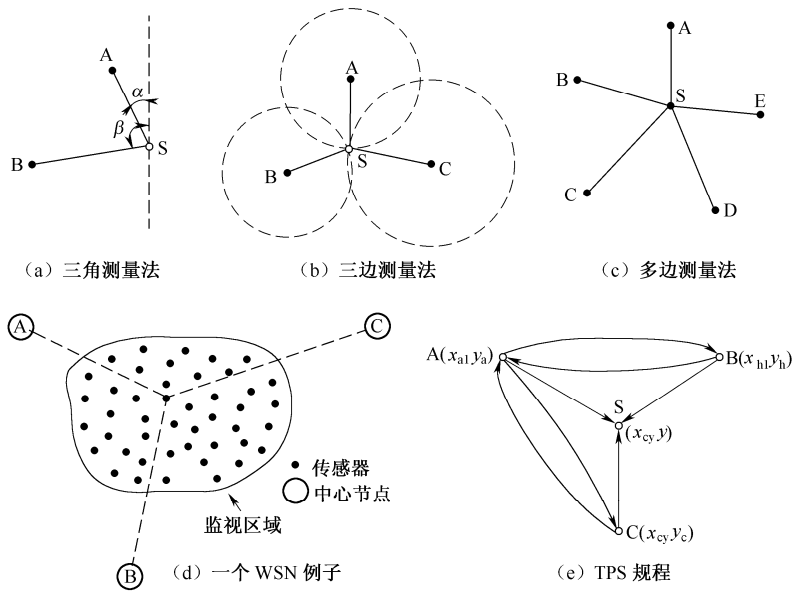


图 7-1 定位技术

7.1.3 TPS网络模型

假定传感器随机布置在一个地面二维监控区域内。每个传感器具有有限资源（带宽、电

池、CPU、存储器等)，配备一副全向天线。已知 3 个中心节点 A、B、C 的位置分别为 (x_a, y_a) 、 (x_b, y_b) 、 (x_c, y_c) ，A、B、C 位于监控区域边界之外，如图 7-1 (d) 所示。假定 A 是主中心节点，监控区域封闭在角度 $\angle BAC$ 内。有待 TPS 确定的传感器位置为 (x, y) 。每个中心节点的发送能够直达监控区域内的所有传感器。要求三个中心节点不能布置在同一条直线上，否则不能辨别传感器的位置。

A、B、C 周期性发送 RF 信标信号，利用位置寻找辅助每个传感器。A、B、C 具有长期工作电源，能够相互接收对方发送的 RF 信号。A、B、C 之间没有时间同步。要求 A、B、C 精确检测信号到达时间以及精确计算总的往返时延。往返时延包括随机时延与已知系统发送时延和接收时延之和。

假如监控区域很大而使得使用三个中心节点不能覆盖整个区域，那么总是将监控区域划分成若干个较小的子区域，布置更多中心节点。

7.1.4 TPS定位方案

TPS 包含两个步骤：第一步检测三个中心节点 A、B、C 的信号到达时间差，将时间差转换成从传感器到达中心节点的距离差；第二步执行三边测量法，将距离估计转换成坐标。下面将进行详细描述。

1. 基于时间的位置检测方法

已知三个中心节点 A、B、C 的位置分别为 (x_a, y_a) 、 (x_b, y_b) 、 (x_c, y_c) ，确定传感器 S 的位置 (x, y) ，如图 7-1 (e) 所示。因此，A、B 之间的距离 $d_{ab} = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$ ，A、C 之间的距离 $d_{ac} = \sqrt{(x_a - x_c)^2 + (y_a - y_c)^2}$ 。S 到达 A、B、C 的距离分别表示为 d_{sa} 、 d_{sb} 、 d_{sc} ，均是待求的距离。在图 7-1 (e) 中，传感器 S 就地测量中心节点 A、B、C 发送的信标信号的 TDoA，接收 B、C 发送的往返时延信息。B 接收到 A 发送的信标信号后就开始发送自己的信标信号，C 接收到 A 和 B 发送的信标信号后才开始发送自己的信标信号。按照周期 T 秒重复这个规程。TPS 进行以下两步操作。

第一步：距离检测。

设 A 是主中心节点，按周期 T 秒发送信标信号。每当 A 发送一个信标信号时就开始一个信标间隔时间。考虑任意信标间隔 i ，传感器 S 以及中心节点 B、C 分别在时刻 t_1^i 、 t_b^i 、 t_c^i 接收到 A 发送的信标信号。B 在时刻 $t_b^{i'}$ ($\geq t_b^i$) 回送 A 一个信标信号，该信标信号包含时间差信息 $t_b^{i'} - t_b^i = \Delta t_b^i$ ，并且在时刻 t_2^i 传播到达 S。C 接收到 A、B 发送的信标信号后，在时刻 t_c^i 回送 A 一个信标信号，该信标信号包含时间差信息 $t_c^{i'} - t_c^i = \Delta t_c^i$ ，并且在时刻 t_3^i 传播到达 S。根据三角不等式， $t_1^i < t_2^i < t_3^i$ 。令 $\Delta t_1^i = t_2^i - t_1^i$ ， $\Delta t_2^i = t_3^i - t_1^i$ ，于是得到

$$d_{ab} + d_{sb} - d_{sa} + v \times \Delta t_b^i = v \times \Delta t_1^i \quad (7-1)$$

$$d_{ac} + d_{sc} - d_{sa} + v \times \Delta t_c^i = v \times \Delta t_2^i \quad (7-2)$$

于是

$$d_{sb} = d_{sa} + v \times \Delta t_1^i - d_{ab} - v \times \Delta t_b^i = d_{sa} + k_1^i \quad (7-3)$$

$$d_{sc} = d_{sa} + v \times \Delta t_2^i - d_{ac} - v \times \Delta t_c^i = d_{sa} + k_2^i \quad (7-4)$$

式中, d_{sa} 、 d_{sb} 、 d_{sc} 为正实数, 且

$$k_1^i = v \times \Delta t_1^i - v \times \Delta t_b^i - d_{ab} \quad (7-5)$$

$$k_2^i = v \times \Delta t_2^i - v \times \Delta t_c^i - d_{ac} \quad (7-6)$$

求 I 个信标间隔的 k_1^i 、 k_2^i 的平均值

$$k_1 = \frac{v}{I} \left[\sum_{i=1}^I (\Delta t_1^i - \Delta t_b^i) \right] - d_{ab} \quad (7-7)$$

$$k_2 = \frac{v}{I} \left[\sum_{i=1}^I (\Delta t_2^i - \Delta t_c^i) \right] - d_{ac} \quad (7-8)$$

下一步运用三边法以及 k_1 和 k_2 计算传感器 S 的坐标 (x, y) 。

说明: ①所有到达时间都是根据本地时钟测量的, 即 t_1 、 t_2 、 t_3 是根据 S 的本地定时器测量的, t_b 、 t'_b 是根据 B 的本地定时器和已知系统时延测量的, t_c 、 t'_c 是根据 C 的本地定时器和已知系统时延测量的, 没有全网同步时间。②要求主中心节点 A 周期性发送信标信号有两个理由。第一, 求多个信标间隔的 k_1^i 、 k_2^i 的平均值有助于减小测量误差, 信标间隔数量 I 能够平衡功耗与可能提高的测量精度。第二, 传感器可以休眠来节省能量, 或者可以在不同时间被使用, 或者可以在其寿命有效期间四处移动。A 周期性发送的信标信号以及 B 和 C 的应答信号有助于随时进行位置寻找。

第二步: 位置计算。

根据式 (7-3)、式 (7-4)、式 (7-7) 和式 (7-8) 可得

$$d_{sb} = d_{sa} + k_1 \quad (7-9)$$

$$d_{sc} = d_{sa} + k_2 \quad (7-10)$$

根据三边法, 得到三个未知变量 x 、 y 、 d_{sa} ($d_{sa} > 0$) 的三个等式如下:

$$(x - x_a)^2 + (y - y_a)^2 = d_{sa}^2 \quad (7-11)$$

$$(x - x_b)^2 + (y - y_b)^2 = (d_{sa} + k_1)^2 \quad (7-12)$$

$$(x - x_c)^2 + (y - y_c)^2 = (d_{sa} + k_2)^2 \quad (7-13)$$

下面介绍 x 、 y 、 d_{sa} 的有效计算方法, 给出唯一解的条件。

2. 采用三边测量法的高效位置检测

不失一般性, 假设三个中心节点 A、B、C 分别位于 $(0,0)$ 、 $(x_1,0)$ 、 (x_2,y_2) , $x_1 > 0$, $y_2 > 0$, 即 $x_a = y_a = y_b = 0$, $x_b = x_1$, $x_c = x_2$, $y_c = y_2$ 。设传感器 S 位于 (x,y) 。通过旋转和变换总是能够将真实位置转换成这种坐标系。下面计算 S 的坐标位置。

根据上述假设条件和式 (7-11)、(7-12)、(7-13) 得到

$$x^2 + y^2 = d_{sa}^2 \quad (7-14)$$

$$x^2 - 2xx_1 + x_1^2 + y^2 = d_{sa}^2 + 2d_{sa}k_1 + k_1^2 \quad (7-15)$$

$$x^2 - 2xx_2 + x_2^2 + y^2 - 2yy_2 + y_2^2 = d_{sa}^2 + 2d_{sa}k_2 + k_2^2 \quad (7-16)$$

联合解方程组式 (7-14)、式 (7-15)、式 (7-16) 可得

$$x = \frac{-2k_1d_{sa} - k_1^2 + x_1^2}{2x_1} \quad (7-17)$$

$$y = \frac{(2k_1x_2 - 2k_2x_1)d_{sa}}{2x_1y_2} + \frac{k_1^2x_2 - k_2^2x_1 + x_2^2x_1 + y_2^2x_1 - x_1^2x_2}{2x_1y_2} \quad (7-18)$$

将式 (7-17)、式 (7-18) 代入式 (7-14) 可得

$$\alpha d_{sa}^2 + \beta d_{sa} + \gamma = 0 \quad (7-19)$$

式中

$$\alpha = 4 \left[k_1^2 y_2^2 + (k_1 x_2 - k_2 x_1)^2 - x_1^2 y_2^2 \right] \quad (7-20)$$

$$\beta = 4 \left[k_1 (k_1^2 - x_1^2) y_2^2 + (k_1 x_2 - k_2 x_1) (k_1^2 x_2 - k_2^2 x_1 + x_2^2 x_1 + y_2^2 x_1 - x_1^2 x_2) \right] \quad (7-21)$$

$$\gamma = (k_1^2 - x_1^2) y_2^2 + (k_1^2 x_2 - k_2^2 x_1 + x_2^2 x_1 + y_2^2 x_1 - x_1^2 x_2)^2 \quad (7-22)$$

定理 7-1: 当且仅当满足以下三个条件之一时, 式 (7-19) d_{sa} 有唯一正数解。

① $\alpha=0, \beta<0, \gamma>0$;

② $\alpha\beta<0$;

③ $\alpha\beta<0, \gamma=\beta^2/4\alpha$ 。

证明: 采用个案分析如下。首先分析 $\alpha=0, \beta=0$: 若 $\gamma=0$, 则式 (7-19) 成立; 若 $\gamma \neq 0$, 则式 (7-19) 不成立。

接着分析 $\alpha=0, \beta \neq 0$: 式 (7-19) 存在唯一根 $d_{sa} = -\gamma/\beta$ 。因为 $\gamma \geq 0$, 所以当且仅当 $\beta < 0$ 和 $\gamma > 0$ 时, $-\gamma/\beta$ 才是正数, 这符合定理 7-1 的条件①。

下面分析 $\alpha \neq 0$ 。

若 $\alpha\gamma < 0$: 则意味着 $\gamma > 0, \alpha < 0$, 进而有 $\beta^2 - 4\alpha\gamma > \beta^2$, 所以式 (7-19) 存在唯一正根 $d_{sa} = \frac{-\beta - \sqrt{\beta^2 - 4\alpha\gamma}}{2\alpha}$ 。这符合定理 7-1 的条件②。

若 $4\alpha\gamma > \beta^2$, 则式 (7-19) 不存在实数根。

若 $0 < \alpha\gamma < \beta^2$, 则式 (7-19) 存在两个相同符号的根 $d_{sa}^{(1)} = \frac{-\beta + \sqrt{\beta^2 - 4\alpha\gamma}}{2\alpha}$ 、
 $d_{sa}^{(2)} = \frac{-\beta - \sqrt{\beta^2 - 4\alpha\gamma}}{2\alpha}$ 。当 $4\alpha\gamma = \beta^2$ 时, 当且仅当 $\beta < 0$ 时式 (7-19) 存在唯一正数根 $d_{sa} = -\beta/2\alpha$ 。

这符合定理 7-1 的条件③。

若 $\gamma = 0$, 则意味着 $k_1^2 = x_1^2$, 因此 $k_1^2 x_2 - k_2^2 x_1 + y_2^2 x_1 - x_1^2 x_2 = 0$ 。所以 $\gamma = 0$, 则 $\beta = 0$ 。因此式 (7-19) 不存在正数根。

将 d_{sa} 代入式 (7-17) 和式 (7-18), 可得到 S 的坐标 x 和 y 。

在上述方法中运用了平方根函数。计算一个正数 N 的平方根 X 只需要按照如下形式重复 Newton 法少数几次就可完成: $X = 0.5 \times (X + N/X)$ 。仿真实验结果表明重复 4 次 Newton 法就能够得到精确结果。

说明: ① Newton 法二次收敛, 因此用于快速解三边函数。② TPS 有一个重要优点: 通过改进第一步提高性能——求多个信标间隔上的平均时间差只涉及简单的代数计算。采用流行的改进策略 (如最大似然、最小均方) 计算比较复杂。

需要注意的是: 收集的数据可能存在误差。在解线性方程组[见式 (7-17) 和式 (7-18)]时, 当条件数 (线性方程组的条件数定义为最大特征值与最小特征值之比率) 小时, 精度较高。线性方程组式 (7-17) 和式 (7-18) 的条件数等于 $\max\{x_1/y_2, y_2/x_1\}$ 。在设计这种线性系

统时,最好选择中心节点的位置,使比率 x_1/y_2 尽量接近 1。 x_2 不影响系统的条件数。实际上,中心节点可以位于等边三角形的顶点,此时条件数等于 1.155,系统非常稳定。

为了确保 d_{sa} 的唯一正数解,需要满足条件 $\alpha\gamma < 0$ 。根据式 (7-22), $\gamma > 0$ 。因此,充分条件降为 $\alpha < 0$, 即 $k_1^2 y_2^2 + (k_1 x_2 - k_2 x_1)^2 < x_1^2 y_2^2$, 改成如下形式

$$k_1^2 y_2^2 + k_1^2 x_2^2 + k_2^2 x_1^2 - 2k_1 k_2 x_1 x_2 < x_1^2 y_2^2 \quad (7-23)$$

在仿真中,无论传感器远离中心节点还是处在非常接近中心节点的位置,均满足上述条件。若传感器位于中心节点附近(在三角形内),则 d_{sa} 存在两个正数解。若传感器位于三角形内部,则 d_{sa} 的正数解是 $d_{sa}^{(2)} = \frac{-\beta - \sqrt{\beta^2 - 4\alpha\gamma}}{2\alpha}$ 。

7.1.5 TPS技术性能分析

根据 k_1 和 k_2 的测量结果,由三边测量法的方程式 (7-11)、式 (7-12)、式 (7-13) 确定传感器 S 的坐标位置 (x,y) 。 k_1 和 k_2 的测量结果不准确引起传感器位置误差。根据式 (7-7) 和式 (7-8), k_1 和 k_2 是 I 个信标间隔的平均值,而根据中心极限定理,当 I 很大时, k_1 和 k_2 通常具有近似分布。因此,不失一般性,假定 k_1 和 k_2 的分布率分别为正态分布 $N(\mu_1, \sigma_1^2)$ 和 $N(\mu_2, \sigma_2^2)$ 。

1. 理论误差分析

为了简单起见,考虑中心节点 A、B、C 分别位于 $(0,0)$ 、 $(R,0)$ 、 $(0,R)$ 点上。这种中心节点布置符合定理 7-1 的条件①。又为了简化分析,考虑传感器 S 与 A、B、C 的距离。一般情况采用类似方法分析。

假定 $\mu_1 = \mu_2 = 0$ 、 $k_1/R \approx 0$ 、 $k_2/R \approx 0$ 是合理的。为了简化分析,又假定 k_1 和 k_2 相互独立(理论上,可以引入 k_1 和 k_2 之间的相关系数)。将 $x_1 = R$ 、 $x_2 = 0$ 、 $y_2 = R$ 代入式 (7-19), 以及 $k_1^2/R^2 \approx 0$ 、 $k_2^2/R^2 \approx 0$, 得到

$$d_{sa} \approx \frac{\sqrt{2R^2 + 2k_1 k_2} - (k_1 + k_2)}{2} \quad (7-24)$$

将式 (7-24) 代入式 (7-17) 可得

$$x \approx \frac{R}{2} + \frac{k_1 k_2}{2R} - \frac{k_1}{2} \sqrt{2 + \frac{2k_1 k_2}{R^2}} \quad (7-25)$$

因为 $\frac{k_1 k_2}{R^2} = \frac{k_1}{R} \frac{k_2}{R} \approx 0$, 所以

$$x \approx \frac{R}{2} + \frac{k_1 k_2}{2R} - k_1 \sqrt{\frac{1}{2}} = \frac{R}{2} + k_1 k_2^* \quad (7-26)$$

式中, $k_2^* = \frac{k_2}{2R} - \sqrt{\frac{1}{2}}$ 。同理, 由式 (7-18) 可得

$$y \approx \frac{R}{2} + \frac{k_1 k_2}{2R} - k_2 \sqrt{\frac{1}{2}} = \frac{R}{2} + k_2 k_1^* \quad (7-27)$$

式中, $k_1^* = \frac{k_1}{2R} - \sqrt{\frac{1}{2}}$ 。

因为 (x,y) 用于估计传感器 S 的位置,所以需要分析估计误差。存在若干种分析方法,下面介绍一种常用方法:计算每个变量的方差,将方差或者标准偏差作为估计误差测量。

因为 k_1 服从正态分布 $N(\mu_1, \sigma_1^2)$, k_2 服从正态分布 $N(\mu_2, \sigma_2^2)$, 所以线性组合 k_1^* 也服从正态分布 $N(\frac{\mu_1}{2R} - \sqrt{\frac{1}{2}}, \frac{\sigma_1^2}{4R^2})$, 线性组合 k_2^* 服从正态分布 $N(\frac{\mu_2}{2R} - \sqrt{\frac{1}{2}}, \frac{\sigma_2^2}{4R^2})$ 。随机变量 X 的均值表示为 $E(X)$ 、方差表示为 $V(X)$ 。于是,由式(7-26)得到

$$V(x) \approx V(k_1 k_2^*) = E(k_1 k_2^*)^2 - [E(k_1 k_2^*)]^2 = E[k_1^2 (k_2^*)^2] - [E(k_1 k_2^*)]^2 \quad (7-28)$$

因为 k_1 和 k_2 相互独立,所以

$$E(k_1 k_2^*) = E(k_1) E(k_2^*) \quad (7-29)$$

$$E[k_1^2 (k_2^*)^2] = E(k_1^2) E[(k_2^*)^2] = [V(k_1) + (E(k_1))^2] [V(k_2^*) + (E(k_2^*))^2] \quad (7-30)$$

将式(7-29)、式(7-30)代入式(7-28)可得

$$\begin{aligned} V(x) &\approx V(k_1) [E(k_2^*)]^2 + V(k_2^*) [E(k_1)]^2 + V(k_1) V(k_2^*) \\ &= \sigma_1^2 \left(\frac{\mu_2}{2R} - \sqrt{\frac{1}{2}} \right)^2 + \frac{\sigma_2^2}{4R^2} \mu_1^2 + \sigma_1^2 \frac{\sigma_2^2}{4R^2} = \frac{\sigma_1^2 \mu_2^2 + \sigma_2^2 \mu_1^2 + \sigma_1^2 \sigma_2^2}{4R^2} + \sigma_1^2 \left(\frac{1}{2} - \frac{\mu_2}{R} \sqrt{\frac{1}{2}} \right) \end{aligned}$$

因为 $\mu_1 = \mu_2 = 0$, 所以

$$V(x) \approx \frac{\sigma_1^2}{2} + \frac{\sigma_1^2 \sigma_2^2}{4R^2} = \frac{\sigma_1^2}{2} \left(1 + \frac{\sigma_2^2}{2R^2} \right) \quad (7-31)$$

同理得到

$$\begin{aligned} V(y) &\approx V(k_1^*) [E(k_2)]^2 + V(k_2) [E(k_1^*)]^2 + V(k_2) V(k_1^*) \\ &= \frac{\sigma_1^2 \mu_2^2 + \sigma_2^2 \mu_1^2 + \sigma_1^2 \sigma_2^2}{4R^2} + \sigma_2^2 \left(\frac{1}{2} - \frac{\mu_1}{R} \sqrt{\frac{1}{2}} \right) \\ &\approx \frac{\sigma_2^2}{2} + \frac{\sigma_1^2 \sigma_2^2}{4R^2} = \frac{\sigma_2^2}{2} \left(1 + \frac{\sigma_1^2}{2R^2} \right) \end{aligned} \quad (7-32)$$

根据上述分析,得到如下结果:① 随机变量 x 、 y 的方差均与 k_1 和 k_2 有关;② k_1 的方差对 x 的方差影响强于 k_2 的方差, k_2 的方差对 y 的方差影响强于 k_1 的方差;③当 R 取大值时, $V(x) \approx \sigma_1^2/2$, $V(y) \approx \sigma_2^2/2$, 证明 x 的方差依赖 k_1 的方差, y 的方差依赖 k_2 的方差;④若 $\sigma_1^2 = \sigma_2^2$, 则 x 、 y 的方差相同。

考虑到 k_1 和 k_2 的加性正态假设条件,能够得到 x 和 y 的近似分布。例如,由于 k_1 和 k_2 相互独立,所以 x 的累积分布函数(Cumulative Distribution Function, CDF) $P(x \leq \alpha)$ 近似表达为(α 为任意实数) $\iint_{\xi\eta, \alpha-R/2} \frac{R}{\pi\sigma_1\sigma_2} \exp\left\{-\frac{1}{2}f(\xi,\eta)\right\} d\xi d\eta$

式中

$$f(\xi, \eta) = \left(\frac{\xi - \mu_1}{\sigma_1} \right)^2 + \left(\frac{2R\eta - \mu_2 + \sqrt{2R^2}}{\sigma_2} \right)^2$$

$$= \left(\frac{\xi}{\sigma_1} \right)^2 + R^2 \left(\frac{2\eta + \sqrt{2}}{\sigma_2} \right)^2 = \left(\frac{\xi}{\sigma_1} \right)^2 + 2R^2 \left(\frac{1 + \sqrt{2}\eta}{\sigma_2} \right)^2 \quad (7-33)$$

上述分析结果有助于解释后面的仿真结果。在仿真中，传感器的 TDoA 测量误差服从正态分布或者均匀分布。TDoA 测量结果的方差决定 k_1 和 k_2 的方差。仿真结果说明位置误差与 TDoA 测量结果的方差有密切关系。

2. 误差源

有三个主要误差源影响 TPS，即接收机系统时延、无线多径衰落信道和 NLOS 传输。接收机系统时延是从信号到达接收机天线时刻开始，直到该信号被接收机准确解码时刻为止的这段时间，主要由接收机电子装置和电路决定。当接收机和信道不存在干扰时，接收机系统时延常常是恒定的，或者变化甚小。接收机系统时延可以预先确定，并用来校准测量结果。例如，中心节点 B 和 C 总是可以分别将 Δt_b^i 和 Δt_c^i 减去接收机系统时延后的结果封装到对 A 发送信标信号的应答消息中，从而实现将时间差发送给传感器。同时，传感器就地测量时间差 Δt_1^i 和 Δt_2^i ，所以可能排除接收机系统时延的影响。因此，在 TPS 模型中，假如中心节点 B、C 能够事先提供有关接收机系统时延的精确信息，那么接收机系统时延对位置检测的影响可忽略不计。

无线多径衰落信道对任何位置检测系统的定位精度均有极大影响。影响多径衰落的主要因素有多径传播、接收机移动速度、周围目标的移动速度、传输信号带宽。多径传播就是发送信号沿着多条传播路径到达接收机天线。在 TPS 模型中，所有传感器和中心节点固定不动，所以 TPS 位置检测精度不受接收机移动速度的影响，但是周围环境中的移动目标（比如汽车、坦克等）可能会产生干扰。

多径信号有两个重要特征。第一个特征是，多条非直线路径的信号传输路径较长，因而总是迟后直线路径信号到达接收机天线；第二个特征是，在 LOS 传输模型中，非直线路径信号在扩散中会损失一些信号功率，因而通常弱于直线路径信号。假如存在 NLOS，那么非直线路径信号可能较强，这是因为直线路径信号在某些方面可能被阻挡。根据这些特征，科学家就能够设计更加灵敏的接收机，锁定和跟踪直线路径信号。例如，对于高精度 DS-BPSK 接收机，运用伪随机码的多径信号迟后直线路径信号到达接收机的影响可忽略不计。TPS 通过测量多个信标间隔上的 TDoA 来减轻多径衰落的影响。在衰落信道上测量 TDoA 非常有效，因为已经能够排除多径衰落造成的许多不利影响以及处理时延。

跟无线信道有关，造成位置检测误差的另一个因素是 NLOS 传输。为了减轻 NLOS 影响，可以将中心节点精心布置在目标周围，使所有中心节点之间、中心节点与传感器之间进行视距通信。

下面将研究 TPS 在衰落信道上的性能，只考虑在传感器上测试的 TDoA 信息是不准确的。所考虑的误差源包括多径衰落和 NLOS。假定 TDoA 测量符合正态分布或者均匀分布。这些假设条件在测试衰落信道上的 TDoA 的研究文献中很常用。

7.2 贪婪地理路由算法

地理路由适用于 WSN。WSN 的通信常常通过物理位置进行寻址。例如，用户不是查询

具有特定 ID 的某个传感器，而是常常查询某个地理区域，位于地理区域内的传感器的 ID 并不重要，地理区域内接收到用户查询消息的任何传感器都可以参与数据累积，并向用户报告结果。这种位置中心通信方式可以采用地理路由，但是没有位置目录服务带来的开销。地理路由根据本地状态（比如一跳相邻节点的位置）作出高效路由决策。地理路由的这种局部特性使其具有良好的可扩展性，因此适用于大规模分布式微型感知应用。

作为最简单的地理路由，贪婪转发（Greedy Forwarding, GF）对 WSN 特别有吸引力。这里将 GF 作为一种简单路由协议：节点总是将分组转发给离目的节点最近的相邻节点。不同的算法可能采用不同的距离定义（比如欧几里得距离、到达目的节点的直线传输距离）。GF 开销低而易于在资源有限的 WSN 平台上实现。但是在随机网络图中，GF 常常由于路由空白问题而操作失败。稍后将证明 GF 对于布置在凸面区域中的感知覆盖网络是一个可行且有效的路由协议。

7.2.1 概述

作为最简单的地理路由，GF 只是根据一条相邻节点的位置信息作出路由决策，因而避免了维护全网拓扑信息带来的开销。GF 总是选择特定路由参数最小化的下一个转发跳节点。GF 的路由参数有若干个，其中包括到达目的节点的欧几里得距离、到达目的节点的直线传输距离（当前节点与目的节点之间的直线传输距离）、到达目的节点的方向（当前节点与目的节点之间的连接直线、当前节点的一个相邻节点与目的节点之间的连接直线之间的角度）。但是，当一个节点不能找到优于自己的相邻节点的时候，假如该节点遭遇局部极小化问题，那么 GF 就可能失败，并且这种局部极小化问题在 Ad Hoc 网络中非常常见。当遭遇局部极小化问题时，地理路由算法就必须从中恢复过来，但是恢复操作必然要引入额外开销和增加复杂性。

1. 假设条件

假设所有传感器节点位于一个二维空间中，每个传感器节点具有相同感知距离 R_S 。对于位于点 p 的一个节点，使用圆 (p, R_S) 表示该节点的感知圆（即感知范围为一个圆），圆的半径为 R_S ，圆心在点 p 。一个节点能够覆盖其感知圆内的任意点。假定一个节点不覆盖其感知圆周上的点。尽管在实际中这个假设对 WSN 的性能影响极小，但是却简化了所做的理论分析。假定 WSN 的展开区域是凸面区域。假如一个 WSN 的展开区域中的任意点至少受到一个节点的覆盖，则称该 WSN 是感知覆盖的。

假定首先采用确定性通信模型进行分析。在该模型中，任意两个节点 u 和 v 当且仅当满足欧几里得距离 $|uv| \leq R_C$ 时才能够直接相互通信， R_C 表示网络的通信距离。在这个通信模型下，可以将一个网络表示为一个单位圆图 $G(V, E)$ ， V 表示网络中的节点集合， E 表示网络中的边集合，当且仅当 $|uv| \leq R_C$ 时边 $(u, v) \in E$ 。然后将基于确定性通信模型的算法和分析扩充到概率通信模型中，注意后者考虑了不可靠 WSN 的特点。

2. 两倍距离特性

通信距离 R_C 与感知距离 R_S 之比 R_C/R_S 对感知覆盖网络的可实现路由质量具有重要影响。

将 R_C/R_S 称为距离比率。直观上，随着距离比率的增大，感知覆盖网络将变得越来越密集，路由质量越来越好。

在实际中，通信距离和感知距离都与系统平台、应用、环境密切相关。一个无线网络接口的通信距离取决于电台属性（比如发射功率、基带/宽带、天线等）和环境（比如室内、室外）。表 7-1 给出了几种无线（传感器）网络接口的室外无线传输距离。经验研究表明：Mica1 Mote 的有效无线传输距离随着环境的不同而变化，通常小于技术说明书描述的距离。

WSN 的感知距离依赖传感器特征、传感器设计以及特定感知应用的要求。感知距离对感知应用的性能具有重要影响，通常根据经验来确定，以便满足应用提出的信噪比（Signal-to-Noise Ratio, SNR）要求。例如，参考文献[16]的经验结果指出：目标分类性能随着传感器与目标间的距离增大而迅速下降。运用 SensoriasGate 公司生产的传感器平台 sGate 作如下实验：在一个区域布置该传感器，各种不同类型军用汽车驶过该区域，根据声学测试识别汽车的类型。实验结果指出：随着传感器-目标间的距离增大，正确识别汽车的概率迅速下降；当传感器-目标间的距离大于 100 m 后，正确识别汽车的概率低于 50%。因此，有效感知距离远小于 100 m。参考文献[19]介绍的类似应用实验结果指出：地震传感器的感知距离大约是 50 m。

显然，由于 WSN 的多种多样，距离比率变化范围大。这里重点在于两倍距离特性（即 $R_C/R_S \geq 2$ ）的 WSN。这个假设条件受到参考文献[27]介绍的几何分析的启示，其分析结果指出：感知覆盖网络若是具有两倍距离特性，则总是连通的。因为网络连通性对于任何路由算法寻找路由是必需的，所以以两倍距离特性作为起点是合理的。

实际经验指出：两倍距离特性适用于许多代表性感知应用。例如，前面提到的基于 sGate 的网络用于目标分类^[16]，其感知距离 $R_S < 100$ m、通信距离 $R_C = 1\,640$ 英尺（547 m），如见表 7-1 所示，因此其距离比率 $R_C/R_S > 5.47$ 。假如地震传感器配备通信距离 $R_C \geq 100$ m 的无线网络接口，那么仍然保持两倍距离特性。

下面介绍的所有结果和分析均假定 WSN 布置在凸面区域内，并且具有两倍距离特性，特别说明除外。

表 7-1 几种无线网络平台的无线通信距离

平台	Mica1 Mote	Mica2 Mote	Sensoria	IEEE 802.11b (SonicWall)
R_C / (英尺)	100	1 000	1 640	1 200~2 320

3. 性能参数

可以采用路由路径的网络长度（转发跳数）和欧几里得距离（每个转发跳的欧几里得距离之和）来描述路由算法的性能。最小网络长度路径可能不同于最小欧几里得距离路径。这里重点考虑路径网络距离。网络长度对多跳 Ad Hoc 网络的时延和吞吐量有重要影响。能够找到最小欧几里得长度路径的路由算法通过控制无线节点的发射功率可能降低能耗^[25, 34]。

路由算法的性能固然受到基本网络路径质量的影响。展开因数是比較两个图的路径质量的一个重要性能参数。设 $\tau_G(u,v)$ 、 $d_G(u,v)$ 分别表示图 $G(V,E)$ 中节点 u 和 v 之间的最小网络长度和最小欧几里得距离。若 $\forall u$ 和 v , $\tau_H(u,v) \leq t \times \tau_G(u,v)$, 则子图 $H(V,E')$ 是图 $G(V,E)$ 的网络 t -生成图, $E' \subseteq E$ 。同理, 若 $\forall V$ 以及 u 和 v , $d_H(u,v) \leq t \times d_G(u,v)$, 则子图 $H(V,E')$ 是图 $G(V,E)$ 的欧几里得 t -生成图, 也将 t 叫做生成图 $H(V,E')$ 的（欧几里得）展开因数。

运用膨胀 (Dilation) 表示无线网络 $G(V,E)$ 相对于理想无线网络的展开因数, 理想无线网络中任意两个节点 u 和 v 之间存在一条具有网络长度 $\lceil |uv|/R_C \rceil$ 的路径以及一条具有欧几里得距离 $|uv|$ 的路径。定义如下两个膨胀:

定义 1: 网络 $G(V,E)$ 的网络膨胀定义为 $D_n = \max_{u,v \in V} (\tau_G(u,v) / \lceil |uv|/R_C \rceil)$ 。

定义 2: 网络 $G(V,E)$ 的欧几里得膨胀定义为 $D_e = \max_{u,v \in V} (d_G(u,v) / |uv|)$ 。

在图论中广泛使用欧几里得膨胀来描述图的质量。显然, 一个无线网络的膨胀是关于由相同节点集组成的任意可能无线网络的展开因数的上限。渐进网络膨胀 (用 \tilde{D}_n 表示) 等于网络长度接近无穷大时的网络膨胀汇聚值。 \tilde{D}_n 用来描述大规模无线网络的路径质量。假如定义 1 中的 $\tau_G(u,v)$ 表示由路由算法 R 选择的节点 u 和 v 之间的路由路径的网络长度, 则称 $D_n(R)$ 是无线网络 $G(V,E)$ 在路由算法 R 下的网络膨胀 (或者简称为 R 的网络膨胀)。一个路由算法的网络膨胀描述该算法相对于理想情形 (即任意两个节点 u 和 v 之间的路径长度为 $\lceil |uv|/R_C \rceil$ 个转发跳) 的性能。 R 的欧几里得膨胀的定义与此类似。

7.2.2 基于DT的膨胀分析

下面根据德劳内三角系 (Delaunay Triangulation, DT) 研究感知覆盖网络的膨胀特性。首先说明: 当满足两倍距离特性时, 一个感知覆盖网络的 DT 就是该网络的一个子图。然后对感知覆盖网络的网络碰撞和欧几里得膨胀进行定量分析。

1. Voronoi图与德劳内三角系

对于二维平面中 n 个节点组成的集合 V , V 的 Voronoi 图就是将该平面划分成 n 个 Voronoi 区, V 中每个节点一个 Voronoi 区。节点 i 的 Voronoi 区表示为 $\text{Vor}(i)$, $i \in V$ 。图 7-2 (a) 给出了一个节点集的 Voronoi 图。当且仅当平面上一个点离节点 i 最近时, 该点位于 $\text{Vor}(i)$ 内。两个相邻 Voronoi 区的边界叫做 Voronoi 边。一条 Voronoi 边位于连接两个相邻节点的区域的中垂线上。Voronoi 边的交叉点就是 Voronoi 顶点。如图 7-2 (a) 所示: 点 p 是三个相邻 Voronoi 区 $\text{Vor}(u)$ 、 $\text{Vor}(v)$ 、 $\text{Vor}(w)$ 的 Voronoi 顶点。假定所有节点处于普通位置上, 即同一条直线上不存在三个节点、同一个圆周上不存在四个节点。

在 Voronoi 图的二重图即德劳内三角系 [用 $\text{DT}(V)$ 表示] 中, 当且仅当 $\text{DT}(V)$ 中节点 u 和 v 的 Voronoi 区共享一条边界线, 则 u 和 v 之间存在一条边。 $\text{DT}(V)$ 由德劳内三角系组成。 $\text{DT}(V)$ 是平面, 即不存在交叉的边。参考文献 [15] 已经证明: 德劳内三角系是完整欧几里得图的有效欧几里得生成图, 其欧几里得展开因数上限等于 $\pi(1+\sqrt{5})/2$ 。参考文献 [20] 证明了一个较严格的展开因数上限等于 $4\sqrt{3}\pi/9 \approx 2.42$ 。

2. 膨胀特性

下面研究感知覆盖网络的欧几里得膨胀和网络膨胀, 首先研究感知覆盖网络的 Voronoi 图和 DT 的特性。结果得到感知覆盖网络的有界膨胀。

对于布置在凸面区域 A 中的一个感知覆盖网络, 位于 A 边界线上附近的一个节点的

Voronoi 区可能超出 A 边界线甚至无边际。下面只考虑布置区域 A 上的局部 Voronoi 图及其相应的二重图。如图 7-2 (a) 所示, 局部 Voronoi 图上任意节点的 Voronoi 区均被限制在区域 A 内。因此, 局部 Voronoi 图的二重图也是一个局部 DT, 这个局部 DT 不包含任意两个节点 (其原始 Voronoi 图的 Voronoi 区延伸到区域 A 外) 之间的边。

在感知覆盖凸面区域中, 任意点被其最近节点所覆盖。由此得到如下引理:

引理 7-1 (覆盖引理): 给定一个节点集 V 以及一个凸面区域 A , 当且仅当 V 中的每个节点能够覆盖其 Voronoi 区 (包括边界线), 则该凸面区域 A 被节点集 V 所覆盖。

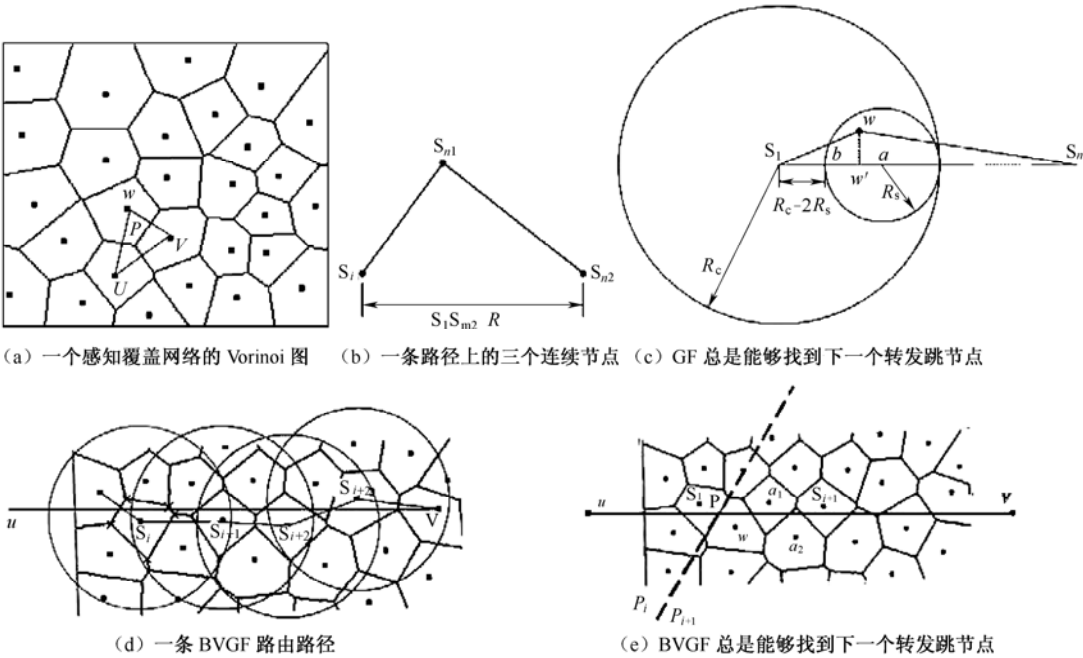


图 7-2 贪婪地里路由说明图

证明: 各个节点将凸面区域 A 分成许多 Voronoi 区。显然, 假如每个 Voronoi 区 (包括边界线) 被其内部节点所覆盖, 那么凸面区域 A 就是感知覆盖的。另一方面, 假如凸面区域 A 被覆盖, 那么 A 中任意点必定被其最近节点所覆盖。在 Voronoi 图中, 一个 Voronoi 区中的所有节点共享同一个最近节点。因此, 每个节点能够覆盖其 Voronoi 区中所有点。两个 Voronoi 区 $\text{Vor}(i)$ 和 $\text{Vor}(j)$ 边界线上的任意点离节点 i 和 j 的距离相等, 被节点 i 和 j 所覆盖。

根据引理 7-1, 一个感知覆盖网络中的每个 Voronoi 区 $\text{Vor}(u)$ 被限制在节点 u 的感知圆内。由这个特性得到如下引理:

引理 7-2: 对于布置在一个凸面区域 A 中的一个感知覆盖网络 $G(V,E)$, 各个节点的德劳内三角系就是该网络的一个子图, 即 $\text{DT}(V) \subseteq G(V,E)$ 。且任意 DT 边的长度小于 $2R_s$ 。

证明: 显然, 两个图 $\text{DT}(V)$ 和 $G(V,E)$ 共享同一个顶点集。现在证明节点 u 和 v 之间的任意 DT 边也是 $G(V,E)$ 中的一条边。如图 7-2 (a) 所示, Voronoi 顶点 p 是三个相邻 Voronoi 区 $\text{Vor}(u)$ 、 $\text{Vor}(v)$ 、 $\text{Vor}(w)$ 的交叉点。根据引理 7-1, p 被节点 u 、 v 、 w 所覆盖。因此, $|pu|$ 、 $|pv|$ 、 $|pw|$ 均小于 R_s 。根据三角形不等式, 有 $|uv| \leq |up| + |pv| < 2R_s$ 。根据两倍距离特性, 有 $|uv| < R_s$ 。因此, uv 是 $G(V,E)$ 中的一条边。

由于感知覆盖网络的单位圆图包含各个节点的 DT，所以感知覆盖网络的膨胀特性至少与 DT 一样。

定理 7-2: 一个感知覆盖网络 $G(V,E)$ 存在欧几里得膨胀 $4\sqrt{3}\pi/9$ ，即 $\forall u, v \in V, d_G(u,v) \leq 4\sqrt{3}\pi|uv|/9$ 。

证明: 根据参考文献[20]的证明，DT 展开因数上限等于 $4\sqrt{3}\pi/9$ 。根据引理 7-2， $DT(V) \subseteq G(V,E)$ ，因此 $\forall u, v \in V, d_G(u,v) \leq d_{DT}(u,v) \leq 4\sqrt{3}\pi|uv|/9$ 。

除了定理 7-2 指出感知覆盖网络具有竞争性的欧几里得膨胀之外，下面定理 7-3 说明感知覆盖网络还具有良好的网络膨胀。

定理 7-3: 在一个感知覆盖网络 $G(V,E)$ 中，节点 u 和 v 之间的最短路径的网络长度满足： $\tau_G(u,v) \leq \left\lfloor (8\pi\sqrt{3}/9) \times (|uv|/R_C) \right\rfloor + 1$ 。

证明: 假如节点 u 和 v 在 $G(V,E)$ 中是相邻节点，那么定理 7-3 显然成立。现在考虑节点 u 和 v 之间的距离至少等于 2 的情形。设 Π 表示 $G(V,E)$ 中节点 u 和 v 之间具有最小欧几里得长度的最短路径， N 表示 Π 的网络长度。考虑 Π 上三个连续节点 S_i, S_{i+1}, S_{i+2} ，如图 7-2 (b) 所示。显然节点 S_i 和 S_{i+2} 在 $G(V,E)$ 中不存在边，否则就会选择节点 S_{i+2} 作为节点 S_i 的下一个转发跳，从而得到一条欧几里得长度比 Π 小的路径，这与 Π 是节点 u 和 v 之间具有最小欧几里得长度的最短路径的假设相矛盾。因此， $|S_i S_{i+2}| > R_C$ 。根据三角形不等式，有 $|S_i S_{i+1}| + |S_{i+1} S_{i+2}| \geq |S_i S_{i+2}| > R_C$ 。对路径 Π 上连续转发跳的这种三角形不等式求和，得到：

$$R_C \lfloor N/2 \rfloor < d_G(u,v) \quad (7-34)$$

根据定理 7-2 可得

$$d_G(u,v) \leq (4\pi\sqrt{3}R_C/9) \times (|uv|/R_C) \quad (7-35)$$

根据式 (7-34) 和式 (7-35)，最小网络长度满足： $\tau_G(u,v) \leq N \leq \left\lfloor (8\pi\sqrt{3}/9) \times (|uv|/R_C) \right\rfloor + 1$ 。

忽略 $\tau_G(u,v)$ 中的常数项 1 和舍入，得到感知覆盖网络的网络膨胀的渐进界限。

推论 7-1: 感知覆盖网络的渐进网络膨胀等于 $8\sqrt{3}\pi/9$ 。

定理 7-2 和推论 7-1 证明感知覆盖网络具有良好的欧几里得膨胀特性和网络膨胀特性。

上述分析只考虑了网络的 DT 子图，忽略了非 DT 边的任何通信边。当 R_C/R_S 取大值时，感知覆盖网络中的一条 DT 边可能明显短于 R_C ，基于膨胀界线的 DT 可能非常保守。下面将证明：当 R_C/R_S 较大时，采用贪婪路由算法（如 GF）能够实现明显较严格的膨胀界线。

7.2.3 贪婪转发 (GF)

首先证明：当满足两倍距离特性时，贪婪转发在感知覆盖网络中总是成功的，因此无需采用复杂恢复方式就能够保证分组的交付。然后进一步推导在 GF 条件下感知覆盖网络的网络膨胀的上限值。在 GF 中可以采用几种不同的路由参数，这里重点在于欧几里得距离和直线传输距离两个路由参数。

定理 7-4: 在一个感知覆盖网络中，GF 总是能够找到其中任意两个节点之间的一条路由路径。而且，在每步转发（到达目的节点的最后一步转发除外）过程中，一个节点总是能够找到一个离目的节点更近（按照欧几里得距离和直线传输距离）的下一个转发跳节点，该转发节点与目的节点之间的距离大于 $R_C - 2R_S$ 。

证明：设 s_n 为目的节点、 s_i 为 GF 路由路径上的源节点或者中间节点，如图 7-2 (c) 所示。若 $|s_i s_n| \leq R_C$ ，则 s_n 一跳可达。若 $|s_i s_n| > R_C$ ，则从 $\overline{s_i s_n}$ 上找一个点 a 使得 $|s_i a| = R_C - R_S$ 。由于 $R_C \geq 2R_S$ ，所以点 a 必定在 s_i 的感知圆外面。由于点 a 被覆盖，所以圆 $C(a, R_S)$ 内必定至少存在一个节点，比如节点 w 。

现在证明：选择 w 作为 s_n 的下一个转发跳，朝 s_n 的传递（按照欧几里得距离和直线距离）大于 $R_C - 2R_S$ 。设点 p 是与节点 s_i 最近的点，且又是 $\overline{s_i s_n}$ 与圆 $C(a, R_S)$ 的交叉点。由于圆 $C(a, R_S)$ 与节点 s_i 的通信圆内切，所以 $|s_i p| = R_C - 2R_S$ 。显然， s_n 与圆 $C(a, R_S)$ 内或者圆周上任意点之间的最大距离等于 $|s_n p|$ 。假设节点 w 在直线段 $\overline{s_i s_n}$ 上的映射点为 w' 。得到： $|s_n s_i| - |s_n w'| \geq |s_n s_i| \geq |s_n w| > |s_i p| = R_C - 2R_S$ 。

根据上述关系可以理解：GF 路由路径的一跳直线距离和欧几里得距离（到达目的节点的最后一跳除外）大于 $R_C - 2R_S$ 。因此，GF 总是能够在任意两个节点之间找到一条路由路径。

定理 7-4 建立了 GF 路由路径每步朝目的节点的传递的下限距离 $R_C - 2R_S$ 。因此，定理 7-5 证明一条 GF 路径的网络长度是上限有界的。

定理 7-5：在一个感知覆盖网络中，GF 总是能够在源节点 u 和目的节点 v 之间找到一条长度不大于 $\lfloor |uv| / (R_C - 2R_S) \rfloor + 1$ 个转发跳的路由路径。

证明：设 N 表示节点 u 和 v 之间的 GF 路由路径的网络长度。该路径上的节点为 $s_0(u)$, $s_1, \dots, s_{n-1}, s_n(v)$ 。根据定理 7-4，有

$$|s_0 s_n| - |s_1 s_n| > R_C - 2R_S$$

$$|s_1 s_n| - |s_2 s_n| > R_C - 2R_S$$

...

$$|s_{n-2} s_n| - |s_{n-1} s_n| > R_C - 2R_S$$

对上述全部不等式求和，得到 $|s_0 s_n| - |s_{n-1} s_n| > (N-1)(R_C - 2R_S)$ 。

已知 $|s_0 s_n| = |uv|$ ，得到 $N < (|uv| - |s_{n-1} s_n|) / (R_C - 2R_S) + 1 < \lfloor |uv| / (R_C - 2R_S) \rfloor + 1$ ，所以 $N \leq \lfloor |uv| / (R_C - 2R_S) \rfloor + 1$ 。

根据定理 7-5 和定义 1，网络 $G(V, E)$ 在 GF 条件下的网络膨胀满足

$$D_n(\text{GF}) \leq \max_{u, v \in V} \left(\left(\frac{|uv|}{R_C - 2R_S} + 1 \right) / \left\lfloor \frac{|uv|}{R_C} \right\rfloor \right) \quad (7-36)$$

忽略式 (7-36) 中的常数项 1 并进行舍入，通过计算可以得到感知覆盖网络在 GF 条件下的网络膨胀的渐进界限。

推论 7-2：感知覆盖网络在 GF 条件下的网络膨胀的渐进界限满足

$$\bar{D}(\text{GF}) \leq \frac{R_C}{R_C - 2R_S} = \frac{R_C / R_S}{(R_C / R_S) - 2} \quad (7-37)$$

根据式 (7-37)，网络膨胀上限随着距离比率 R_C / R_S 的增大而单调递减；当 $R_C / R_S > 4$ ，网络膨胀上限小于 2；当 R_C / R_S 很大时，网络膨胀接近 1。这个结果证明以下直觉是正确的：当通信距离明显大于感知距离时，就网络长度而论，感知覆盖网络接近理想网络。

但是，当 R_C / R_S 接近 2 的时候，式 (7-37) 的 GF 网络膨胀范围迅速增大到无穷大。在定理 7-4 的证明过程中，当 R_C 接近 $2R_S$ 时，转发节点 s_i 可能无穷接近圆 $C(a, R_S)$ 与直线 $\overline{s_i s_n}$ 的交叉点。因此 s_i 可能选择圆 $C(a, R_S)$ 内部的一个相邻节点，这个相邻节点朝目的节点的推进无穷

小,从而得到一条长路径。源节点 u 与目的节点 v 间 GF 路由路径的网络长度在 $O\left(\left(|uv|/R_C\right)^2\right)$ 范围内。根据定义 1 能够理解: 在给定距离比率条件下, 由这个结果不能得到网络膨胀的一个恒定上限值。

7.2.4 有界Voronoi贪婪转发 (BVGF)

尽管 GF 在 $R_C/R_S \square 2$ 时具有满意的网络膨胀范围, 但是当 R_C/R_S 接近 2 时, GF 网络膨胀范围变得非常大。而根据 Voronoi 图的分析却是在 R_C/R_S 接近 2 时得到满意的膨胀范围, 当 $R_C/R_S \square 2$ 时这个膨胀范围较保守。在这两种结果激励下开发出新的路由算法, 即有界 Voronoi 贪婪转发 (Bounded Voronoi Greedy Forwarding, BVGF) 算法。BVGF 通过综合 GF 和 Voronoi 图, 在 $R_C/R_S > 2$ 时得到满意的分析膨胀范围。

1. BVGF算法

BVGF 类似于 GF, 也是一个本地算法, 根据一跳相邻节点位置信息作出贪婪路由决策。当节点 i 需要转发一个分组的时候, 只有假如其某个相邻节点 j 满足条件——连接源节点与目的节点的直线段与 $\text{Vor}(j)$ 相交、或者与 $\text{Vor}(j)$ 的一条边界重合, 那么 j 才符合条件作为下一个转发跳。BVGF 从全部合格相邻节点中选择离目的节点的欧几里得距离最短的那个相邻节点作为下一个转发跳。假如存在多个合格相邻节点并且到达目的节点的欧几里得距离最短且相等, 那么就从这几个相邻节点中随机选择一个作为下一个转发跳。图 7-2 (d) 表示从源节点 u 到达目的节点 v 的 BVGF 路由路径上的 4 个连续节点 (s_i 、 s_{i+1} 、 s_{i+2} 、 s_{i+3})。图中绘出了每个节点的同心圆。从图 7-2 (d) 中可以看到: 路由路径上节点的下一个转发跳在 Voronoi 图中可能与自己不相邻 (比如: 节点 s_i 与节点 s_{i+1} 没有共享 Voronoi 边)。当 $R_C \square R_S$ 时, BVGF 实现的膨胀范围比 DF 范围紧, DF 范围只考虑 DT 边, 不会随着距离比率的变化而变化。

GF 与 BVGF 之间的主要差异是: BVGF 只考虑其 Voronoi 区与源节点和目的节点间连接直线交叉的相邻节点。这个特性使得 BVGF 能够达到较紧的网络膨胀上限值。

在 BVGF 中, 每个节点维护一张邻区表。对于每个一跳相邻节点 j , 邻区表包含 j 的位置信息以及 $\text{Vor}(j)$ 的各个顶点的位置信息。例如, 在图 7-2 (d) 中, 对于一跳相邻节点 s_i , 节点 s_{i+1} 的邻区表包含 s_i 的位置信息以及 $\text{Vor}(s_i)$ 的各个顶点 (标有 “x” 符号) 的位置信息。为了维护邻区表, 每个节点周期性广播一条信标消息, 信标消息包含该节点的位置信息以及该节点的 Voronoi 区的各个顶点的位置信息。由于所有 Voronoi 相邻节点处在自己的通信范围内, 所以每个节点能够根据其相邻节点位置信息计算出自己的 Voronoi 顶点。

假定处于一个节点的通信范围内的节点数在 $O(n)$ 内。位于一个节点的邻区内的 Voronoi 顶点数在 $O(n)$ 内。因此, 信标消息长度和邻区表大小均在 $O(Ln)$ 内, L 表示描述一个位置信息所用的比特数。一个节点计算其所有相邻节点的 Voronoi 图的时间复杂性为 $O(n \log n)$ 。

2. BVGF的网络膨胀

下面分析 BVGF 的网络膨胀。为了简化对从源节点 u 到达目的节点 v 的 BVGF 路由路径的讨论, 假定 u 是最初的数据源节点, 连接 u 和 v 的直线为 x 轴。节点 u 和 v 的 Voronoi 转

发矩形是由点 $(0, R_S)$ 、 $(0, -R_S)$ 、 $(|uv|, -R_S)$ 、 $(|uv|, R_S)$ 定义的矩形。设 $x(a)$ 、 $y(a)$ 分别表示点 a 的 x 轴坐标值、 y 轴坐标值。节点 u 和 v 之间的投影距离定义为其 x 轴坐标值之差。

首先证明 BVGF 能够在感知覆盖网络中任意两个节点之间找到一条路由路径(定理 7-6)。接着证明 BVGF 路由路径总是位于 Voronoi 转发矩形内部。然后推导 BVGF 路径每步的投影距离的下限(引理 7-4); 由于 R_C/R_S 接近 2 时这个下限不严谨, 所以又推导 BVGF 路径连续两步和连续四步的投影距离的下限(引理 7-6 和引理 7-7), 这个下限相对较严谨。最后在定理 7-8 中建立 BVGF 的网络膨胀的渐进范围。

BVGF 在每步选择的下一个转发跳满足条件: 下一个转发跳节点的 Voronoi 区与 x 轴相交, 由此得到一条到达目的节点的、由接近 x 轴的节点组成的路由路径。

定理 7-6: 在一个感知覆盖网络中, BVGF 总是能够在任意两个节点间找到一条路由路径, 而且 BVGF 每步的投影距离是正数。

证明: 如图 7-2 (e) 所示, 节点 s_i 是从源节点 u 到达目的节点 v 的 BVGF 路由路径上的一个中间节点。 x 轴与 $\text{Vor}(s_i)$ 相交、或者与 $\text{Vor}(s_i)$ 的一条边界重合。设 p 是 x 轴与 $\text{Vor}(s_i)$ 相交且离 v 较近(假如 x 轴与 $\text{Vor}(s_i)$ 的一条边界重合, 则选择离 v 最近的那个 $\text{Vor}(s_i)$ 顶点作为 p)。必定存在一个节点 w 使得 $\text{Vor}(s_i)$ 和 $\text{Vor}(w)$ 共享一条 Voronoi 边, 这条边包含 p 且与 x 轴相交。这条 Voronoi 边所在的直线[图 7-2 (e) 中的虚线]定义两个半平面 P_i 和 P_{i+1} , 且 $s_i \in P_i$, $w \in P_{i+1}$ 。根据 Voronoi 图的定义, 由于 $v \in P_{i+1}$, $|wv| < |s_i v|$, 所以对于半平面 P_{i+1} 中任意点离 w 、 s_i 的距离是前者比后者短。此外, 由于 $|s_i w| < 2R_S \leq R_C$ (引理 7-2), 直线段 \overline{uw} 与 $\text{Vor}(w)$ 相交[或者与 $\text{Vor}(w)$ 的一条 Voronoi 边界重合], 所以 w 作为下一个转发跳节点是合格的。因此有: 到达目的节点的每步(最后一步除外)以及 BVGF 总是能够在源节点和目的节点之间找到一条路由路径。

现在证明 BVGF 路由路径每步的投影距离是正数。讨论两种情况。①假如 s_i 选择 w 作为 BVGF 路由路径的下一个转发跳, 那么根据 Voronoi 图的定义, s_i 和 w 分别位于直线段 $\overline{s_i w}$ 的中垂线的左边和右边。因此, $x(s_i) < x(p) < x(w)$, s_i 和 w 之间的投影距离是正数。②假如 s_i 选择 s_{i+1} (s_{i+1} 不同于 w) 作为 BVGF 路由路径的下一个转发跳, 那么可以沿着 x 轴建立一条连续路径, 该路径由节点 s_i 、 $a_0(w)$ 、 a_1 、 \dots 、 a_m 、 s_{i+1} 组成, 其中任意两个相邻节点共享一条与 x 轴相交的 Voronoi 边, 如图 7-2 (e) 所示。类似于①, 能够证明: $x(s_i) < x(a_0) < x(a_1) < \dots < x(a_m) < x(s_{i+1})$ 。因此, BVGF 路由路径上依次连续节点 s_i 和 s_{i+1} 之间的投影距离是正数。

BVGF 总是能够将分组转发到其 Voronoi 区与 x 轴相交的节点。运用这个特性, 以及结合每个 Voronoi 区被限制在一个感知圆范围内(引理 7-1), 就能够证明 BVGF 路由路径上任意节点位于 Voronoi 矩形内。

引理 7-3: 从节点 u 到达节点 v 的 BVGF 路由路径位于节点 u 和 v 的 Voronoi 转发矩形内。

在一个感知覆盖网络中, BVGF 的贪婪特性确保节点从其所有合格相邻节点中选择离目的节点最近的那个相邻节点作为下一个转发跳。而根据引理 7-3, 下一个转发跳节点必须位于 Voronoi 转发矩形内。根据这些结果推导出 BVGF 每步投影距离的下限。

引理 7-4 (单步推进引理): 在一个感知覆盖网络中, 一条 BVGF 路由路径的每步投影距离大于 Δ_1 , $\Delta_1 = \max\left(0, \sqrt{R_C^2 - 2R_C R_S} - R_S\right)$ 。

证明: 如图 7-2 (g) 所示, 节点 s_i 是从源节点 u 到达目的节点 v 的 BVGF 路由路径上的一个中间节点。设点 s'_i 是节点 s_i 在 x 轴上的映射点。根据引理 7-3, $s_i s'_i < R_S$ 。设点 d 是 x 轴

上满足 $|s_i d| = R_C - R_S$ 的点。根据引理 7-1，必定存在节点 w ， w 覆盖点 d ，且 $d \in \text{Vor}(w)$ 。显然， w 在圆 $C(d, R_S)$ 内部。因为 d 在 x 轴上且 $d \in \text{Vor}(w)$ ，所以 x 轴与 $\text{Vor}(w)$ 相交，又因为圆 $C(d, R_S)$ 与节点 s_i 的同心圆内切，所以节点 w 在节点 s_i 的同心圆内部。因此，节点 s_i 至少可以选择节点 w 作为下一个转发跳。设 c 是圆 $C(d, R_S)$ 与 x 轴的交叉点，且离 u 最近，设 w' 是节点 w 在 x 轴上的映射点。节点 s_i 和 x 之间的投影距离为： $|s_i' w'| > |s_i' d| - R_S = \sqrt{|s_i' d|^2 - |s_i' s_i'|^2} - R_S > \sqrt{(R_C - R_S)^2 - R_S^2} - R_S = \sqrt{R_C^2 - 2R_C R_S} - R_S$ 。

当 $R_C/R_S \geq 1 + \sqrt{2}$ ， $\sqrt{R_C^2 - 2R_C R_S} - R_S \geq 0$ 。根据定理 7-6，BVGF 的每步投影距离为正数。因此，每步投影距离的下限等于 $\max\left(0, \sqrt{R_C^2 - 2R_C R_S} - R_S\right)$ 。

引理 7-4 说明 BVGF 路由路径上任意两个节点间的投影距离在 $R_C/R_S \geq 1 + \sqrt{2}$ 时可能接近零。那么在这种情况下是否存在较严谨的下限？考虑 BVGF 路由路径上两个非相邻的节点 i 和 j 。节点 i 和 j 间的欧几里得距离必须大于 R_C ，否则 BVGF 就会选择 j 作为 i 的下一个转发跳，这与假定节点 i 和 j 为非相邻节点相矛盾。将 BVGF 的这个特性称为非相邻推进特性。GF 也有类似的特性。得到如下引理：

引理 7-5（非相邻推进特性）：在一个感知覆盖网络中，一条 BVGF 路由路径上任意两个非相邻节点间的欧几里得距离大于 R_C 。

引理 7-5 综合了 BVGF 路径位于 Voronoi 转发矩形内这个特点，从而直观上得到：BVGF 朝目的节点所作的连续两步投影距离是有下限的。得到如下引理 7-6，由引理 7-6 建立比引理 7-4 严谨的、 R_C/R_S 取小值时的 BVGF 投影距离下限。

引理 7-6（两步推进引理）：在一个感知覆盖网络中，一条 BVGF 路由路径上任意两个非相邻节点 i 和 j 间的投影距离大于 $\Delta_2 = \sqrt{R_C^2 - 4R_S^2}$ 。

考虑到非相邻节点位置的不同情形，可以进一步推导 BVGF 连续四步的投影距离的下限。

引理 7-7（四步推进引理）：在一个感知覆盖网络中，一条 BVGF 路由路径上连续四步的

投影距离大于 Δ_4 ，其中 $\Delta_4 = \begin{cases} \sqrt{R_C^2 - 4R_S^2} & \left(2 \leq \frac{R_C}{R_S} \leq \sqrt{5}\right) \\ \sqrt{4R_C^2 - 16R_S^2} & \left(\frac{R_C}{R_S} > \sqrt{5}\right) \end{cases}$ 。

证明：设 $S_0(u)$ 、 S_1 、 \dots 、 S_{n-1} 、 $S_n(v)$ 表示源节点 u 和目的节点 v 间的 BVGF 路由路径上的连续节点。 S_i 、 S_{i+2} 、 S_{i+4} 是该路径上三个非相邻的节点。不失一般性，设 S_i 位于 x 轴的上边。图 7-3 给出 S_i 、 S_{i+2} 、 S_{i+4} 的所有可能结构（虚框表示 Voronoi 转发矩形）。现在推导 S_i 和 S_{i+4} 间的投影距离的下限。

① 当 S_i 和 S_{i+4} 位于 x 轴的不同边时，如图 7-3（a）和图 7-3（b）所示， S_i 和 S_{i+4} 间的投影距离 δ_{ab} 等于 S_i 和 S_{i+2} 间投影距离与 S_{i+2} 和 S_{i+4} 间投影距离之和。根据引理 7-6，得到： $\delta_{ab} = \sqrt{R_C^2 - R_S^2} + \sqrt{R_C^2 - 4R_S^2}$ 。

② 当 S_i 和 S_{i+4} 位于 x 轴的同一边时，如图 7-3（c）和图 7-3（d）所示，根据引理 7-6， S_i 和 S_{i+4} 间的投影距离大于 $\delta_{cd} = \sqrt{R_C^2 - R_S^2}$ 。另一方面， S_i 和 S_{i+4} 间的投影距离等于 S_i 和 S_{i+2} 间投影距离与 S_{i+2} 和 S_{i+4} 间投影距离之和，即根据图 7-3（c）有 $\delta_c = 2\sqrt{R_C^2 - 4R_S^2}$ ，根据图 7-3（d）

有 $\delta_d=2\sqrt{R_C^2-R_S^2}$ 。由于 $\delta_d>\delta_c$ ，所以当 S_i 和 S_{i+4} 位于 x 轴同一边时其间的投影距离的下限等于 $\max\{\delta_{cd}, \delta_c\}$ 。

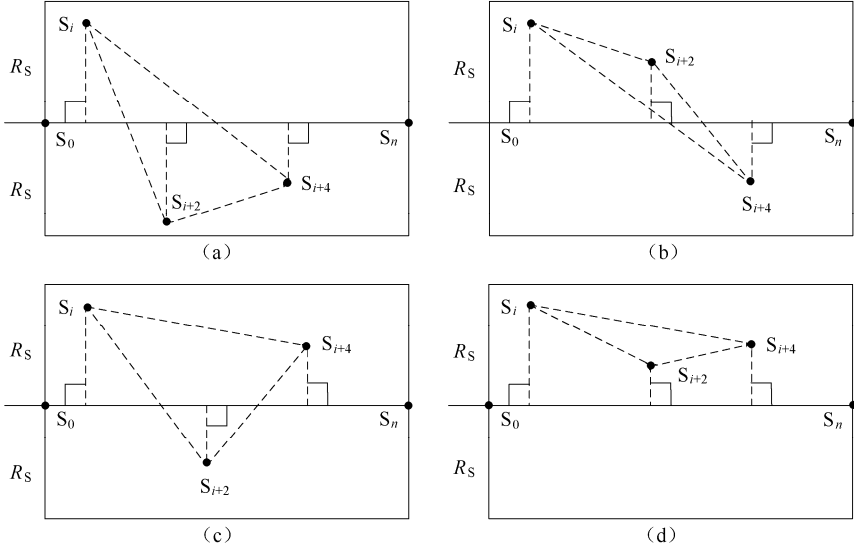


图 7-3 连续四步的投影距离

综合情形①和②，BVGF 路由路径上连续四步的投影距离的下限等于 $\Delta_4=\min\{\delta_{ab}, \max\{\delta_{cd}, \delta_c\}\}$ 。根据 δ_{ab} 、 δ_{cd} 、 δ_c 之间的关系，可以将 Δ_4 转化为引理 7-7 的表达式。

当 R_C/R_S 取小值时，网络相对稀疏。尽管根据引理 7-4 在这种情况下的一步投影距离接近零，但是，引理 7-6 和引理 7-7 却证明 BVGF 朝目的节点的连续两/四步投影距离却是有限的。另外，当 $R_C \square R_S$ 时，网络的感知覆盖范围导致路由节点通信范围内高节点密度，因此 BVGF 每步的投影距离接近 R_C 。在这种情况下，引理 7-4 建立的投影距离下限比引理 7-6 和引理 7-7 建立的投影距离下限严格。

根据引理 7-4、引理 7-6、引理 7-7 得到的一跳、两跳、四跳最小投影距离，就能够推导出 BVGF 路由路径的网络膨胀上限。综合 BVGF 路由路径的网络膨胀上限，得到如下定理 7-7。

定理 7-7: 在一个感知覆盖网络中，任意两个节点 u 和 v 之间的 BVGF 路由路径不会大于 Δ 个转发跳，其中 $\Delta=\min\left\{\left\lceil\frac{|uv|}{\Delta_1}\right\rceil, 2\left\lceil\frac{|uv|}{\Delta_2}\right\rceil+1, 4\left\lceil\frac{|uv|}{\Delta_4}\right\rceil+3\right\}$ 。

通过比较 Δ_1 、 Δ_2 、 Δ_4 （忽略舍入和常数）可以推导出 Δ 的表达式：

- 当距离比率在 $[2, \sqrt{5}]$ 之间时， $\Delta=4/\Delta_4$ ；
- 当距离比率在 $[\sqrt{5}, 3.8]$ 之间时， $\Delta=2/\Delta_2$ ；
- 当距离比率大于 3.8 之间时， $\Delta=1/\Delta_1$ 。

用 Δ 替换定义 1 中的网络长度 $\tau_G(u,v)$ ，可以计算出在 BVGF 条件下的网络膨胀渐进范围 $\tilde{D}_n(\text{BVGF})$ ，即 $\tilde{D}_n(\text{BVGF}) \leq \max_{u,v \in V} (\Delta / \lceil |uv| / R_C \rceil)$ ，所以得到如下定理 7-8。

定理 7-8: 一个感知覆盖网络在 BVGF 条件下的网络膨胀渐进范围满足

$$\tilde{D}_n(\text{BVGF}) \leq \begin{cases} \frac{4R_C}{\sqrt{R_C^2 - R_S^2}} & \left(2 \leq \frac{R_C}{R_S} \leq \sqrt{5} \right) \\ \frac{2R_C}{\sqrt{R_C^2 - 4R_S^2}} & \left(\sqrt{5} < \frac{R_C}{R_S} \leq 3.8 \right) \\ \frac{R_C}{\sqrt{R_C^2 - 2R_C R_S - R_S^2}} & \left(\frac{R_C}{R_S} > 3.8 \right) \end{cases} \quad (7-38)$$

7.2.5 网络膨胀分析总结

下面总结基于确定性通信模型的网络膨胀范围。对于所有大于 2 的距离比率，BVGF 的渐进网络膨胀范围大致相当；当 $R_C/R_S=2$ 时，网络膨胀范围取最差值 $8\sqrt{3}/3 \approx 4.62$ ，因此，当任意两个节点 u 和 v 之间的小于 $4.62 \lceil |uv|/R_C \rceil$ 个转发跳时，BVGF 总是能够找到 u 和 v 之间的一条路由路径。GF 的渐进网络膨胀范围随着距离比率的增大而迅速增大，且当 R_C/R_S 接近 2 时趋于无穷大。当 $R_C/R_S > 3.5$ 时，GF 和 BVGF 的网络膨胀非常类似，这是因为网络拓扑密集，GF 和 BVGF 都能够找到极短的路由路径。当 $R_C/R_S > 2.5$ 时，DT 的网络膨胀范围明显大于 GF 和 BVGF 的网络膨胀范围，这是因为：基于 DT 的分析只考虑 DT 边（在引理 7-2 中已经证明 DT 边的长度小于 $2R_S$ ），当通信距离比感知距离大得多的时候就变得非常保守。

7.2.6 基于概率通信模型的扩充

前面介绍的理论分析和协议设计都是在简化通信模型（即假定确定性通信距离）基础上进行的。但是，真正的传感器网络平台（比如伯克利的 Mote 传感器系列）存在不可靠链路和不规则通信距离。无线传感器网络必须处理不可靠链路引起的通信障碍问题。GF 在有损网络中总是选择离目的节点最近的节点作为下一个转发跳，因此常常得到的是长通信链路而不是不可靠通信链路。下面将上述结果扩充到捕获这些特点的概率通信模型中。

在概率通信模型中，利用分组接收率（Packet Reception Rate, PRR） $[\text{PRR}(u,v)]$ 描述从节点 u 到达节点 v 的通信链路的质量。PRR(u,v) 定义为从 u 到 v 的成功发送数量与从 u 到 v 的总发送数量之比。由于链路通信质量常常不对称，所以 PRR(u,v) 可能不等于 PRR(v,u)。在实际中可以在线或者脱线估计链路的 PRR。例如，在 TinyOS 的 MT 路由协议中，节点通过监视其相邻节点周期性发送的信标消息的接收通信情况，计算其某个相邻节点到达自己的链路 PRR。

1. 具有 ARQ 机制的路由算法

一个节点若是没有成功将其某个分组交付给下一个转发跳（比如没有接收到下一个转发跳节点回送的 ACK），则采用自动重传请求（Automatic Repeat Request, ARQ）机制重传这个分组。假定 ARQ 重传一个分组，直到下一个转发跳节点成功接收到这个分组为止。下面讨论采用 ARQ 后对 GF 和 BVGF 的修改。在有损网络中，对于 GF 就能量效率而论，PRR 与传输距离（到达目的节点的传输距离）之乘积是最佳的。PRR 与传输距离之积平衡转发跳

数和路径可靠性，达到的能量效率优于距离。对 GF 和 BVGF 修改如下，就可以运用 PRR 与传输距离之积这个参数：节点 u 不是从所有合格路由节点中选择离目的节点最近的那个相邻节点作为下一个转发跳，而是选择乘积 $(|ut| - |vt|) \times \text{PRR}(u, v)$ 取最大值的节点 v 作为下一个转发跳，其中 t 表示目的节点。将经过此修改的 GF 和 BVGF 分别表示为 GF_e 和 BVGF_e 。

对前面讨论的两倍距离特性修改如下：对于给定参数 p ($0 < p \leq 1$)，定义概率通信距离 $R_C(p)$ 为其中任意两个节点间的链路具有 $\text{PRR} \geq p$ 的通信距离。扩充两倍距离特性可以表示为 $R_C(p) \geq 2R_S$ 。假如 $\exists p > 0$ ，且 $R_C(p) \geq 2R_S$ ，则 GF_e 和 BVGF_e 总是能够在任意两个节点间找到一条路由路径。引入 $R_C(p)$ 概念只是为了进行性能范围分析。 GF_e 和 BVGF_e 的工作不需要知道 $R_C(p)$ 。

前面 DT、GF、BVGF 的分析重点是转发跳数和网络膨胀。但是在概率通信模型中，由于链路的不可靠，所以转发跳数不能说明路由路径的质量。采用 ARQ 机制后，路由路径的能耗和端到端时延依赖从源节点到目的节点成功交付一个分组所需要的发送总次数。因此，发送总次数是描述路由路径质量的一个较精确的参数。定义如下几个符号：用 $R_C(p)$ 替换式 (7-37) 和式 (7-38) 中的 R_C ， $\tilde{D}_n(\text{GF})$ 和 $\tilde{D}_n(\text{BVGF})$ 表示渐进网络膨胀范围。对于给定路由算法， Δ_i 表示该算法第 i 步朝目的节点的传输距离， p_i 表示该算法第 i 步选择的链路的 PRR， Δ' 表示只考虑 $R_C(p)$ 范围内相邻节点时该算法朝目的节点的最小传输距离。有下列关于 GF_e 和 BVGF_e 性能的定理。

定理 7-9：假设一个感知覆盖网络满足概率通信距离 $R_C(p)$ 的两倍距离特性，采用 ARQ 机制，那么算法 A_e (A 是 GF 或者 BVGF) 在两个节点 u 和 v 间交付一个分组所做的渐进发送总次数的期望值不小于 $\tilde{D}_n(A) \times \frac{|uv|}{p \times R_C(p)}$ 。

证明：根据 $R_C(p)$ 的定义， $R_C(p)$ 内任意链的 PRR 不小于 p 。因为 A_e 选择 PRR 与传输距离乘积取最大值的节点作为下一个转发跳节点，所以得到

$$\forall i, \Delta_i \times p_i \geq \Delta' \times p \quad (7-39)$$

对于 $\text{PRR} = p_i$ 的一条链路，发送次数期望值等于 $1/p_i$ 。根据式 (7-39)，源节点 s 和目的节点 t 间的发送总次数满足如下不等式：

$$\sum_i \frac{1}{p_i} \leq \frac{\sum_i \Delta_i}{\Delta' \times p} = \frac{|st|}{\Delta' \times p} \quad (7-40)$$

根据网络膨胀的定义， $\Delta' = R_C(p) / \tilde{D}_n(A)$ 。替换式 (7-40) 中的 Δ' ，得到定理 7-9 中的表达式。

定理 7-9 说明 GF_e 和 BVGF_e 在有损网络中采用 ARQ 机制后都能够找到发送次数有界的路由路径。

2. 无ARQ机制的路由算法

下面描述在没有 ARQ 机制以及在概率通信模型下对 GF 和 BVGF 的修改。没有采用 ARQ 机制，节点若是没有成功将一个分组交付给下一个转发跳节点，则丢掉这个分组。采用端到端可靠性定量描述路由路径的质量。端到端可靠性定义为一个分组沿着从源节点到达目的节点的路径被成功发送的概率。一条路由路径的端到端可靠性等于该路径上每条链路的 PRR 的乘积。提出一个采用 GF 和 BVGF 算法时的新参数 $(|ut| - |vt|) / \ln[1/\text{PRR}(u, v)]$ ，其中 u 、 v 、

t 都是路由转发节点，分别是 u 和目的节点的一个相邻节点。这个参数提供端到端可靠性的下限。将采用这个参数的 GF 和 BVGF 分别表示为 GF_r 和 $BVGF_r$ 。有下列关于 GF_r 和 $BVGF_r$ 性能的定理。

定理 7-10: 假设一个感知覆盖网络满足概率通信距离 $R_C(p)$ 的两倍距离特性，不采用 ARQ 机制，那么利用算法 A_r (A 是 GF 或者 BVGF) 所建立路径的渐进端到端可靠性不小于

$$\exp\left[\frac{\tilde{D}_n(A) \times |uv| \times \ln p}{R_C(p)}\right]。$$

证明: 由于 A_r 总是选择 $\frac{A_i}{\ln \frac{1}{p_i}}$ 最大的节点作为下一个转发跳节点，所以得到

$$\forall i, \frac{A_i}{\ln \frac{1}{p_i}} \geq \frac{A'}{\ln \frac{1}{p}} \tag{7-41}$$

由式 (7-41) 得到如下不等式

$$\sum_i \ln p_i \geq \frac{\sum_i A_i \times \ln p}{A'} = \frac{|st| \times \ln p}{A'} \tag{7-42}$$

根据式 (7-42)，由 A_r 在源节点 s 和目的节点 t 之间寻找到的路由路径的可靠性满足 $\prod_i p_i \geq e^{\frac{|st| \times \ln p}{A'}}$ 。用 $R_C(p)/\tilde{D}_n(A)$ 替换 A ，根据 $\tilde{D}_n(A)$ 的定义可得到定理 7-10 的表达式。

7.3 位置辅助泛洪协议 (LAF)

位置辅助泛洪 (Location-Aided Flooding, LAF) 是一个 WSN 数据分发协议，采用虚拟栅格概念，将被监视区域 (WSN) 划分成若干个虚拟栅格，然后传感器节点自行决定成为网关节点组和内部节点组。每个传感器节点根据其位置关联一个虚拟栅格。虚拟栅格内的传感器节点分成网关节点和内部节点两种类型，网关节点负责虚拟栅格间的数据转发，内部节点负责在其栅格内部转发数据。LAF 利用传感器节点位置信息减少泛洪固有的冗余传输，节省能量，从而延长 WSN 寿命。

LAF 采用经过改进的经典泛洪，将这个泛洪称为改进泛洪 (Modified Flooding, MF)。下面将描述 MF 的基本思想。

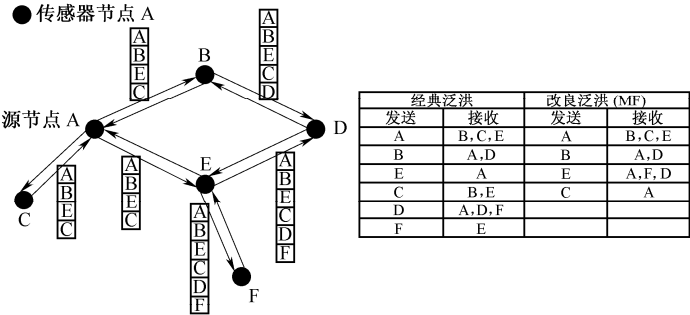
7.3.1 LAF 协议概述

1. 改进泛洪 (MF)

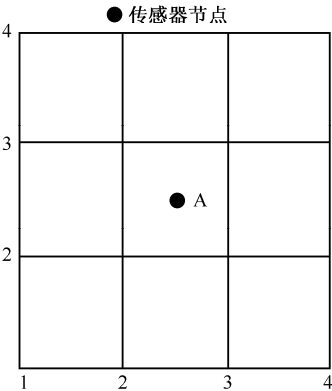
MF 采用传感器节点身份识别码 (ID) 来提高 WSN 中信息分发的能量效率。采用 MF 发送的每个分组的分组头包含一个特定组成域——节点列表。一个分组的节点列表记录已经拥有该分组的所有节点的 ID。假定网络无损耗，那么该分组头信息通知接收节点：该分组节点列表中列出的所有节点已经有了该分组，因此不必再将该分组转发给这些节点。

有两种方法实现 MF。一种方法是采用单目标传输方案，即发送节点只将分组发送给预定的接收节点。但是在无线网络环境下很难实现这种方法。一种比较实用的方法是采用广播机制，允许发送节点的所有相邻节点接收分组。接收节点首先检查所接收分组头节点列表中是否列出其所有相邻节点：假如是，则不广播该分组；假如不是，则广播该分组。接收节点还要检查自己是否在分组头节点列表中：假如已列出，则不处理该分组，丢掉该分组。分组头包含源节点 ID 和序列号信息。这里假定采用后一种 MF 实现方法。

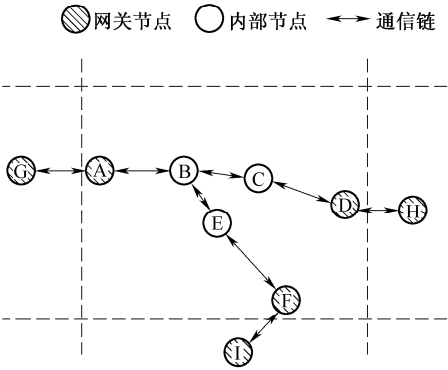
如图 7-4 所示，通过一个例子说明如何使用 MF 减少泛洪中的冗余传输。当节点 S 需要在整个网络中分发数据的时候，S 在数据分组头节点列表中填入自己所有相邻节点的 ID，然后将数据分组广播给自己所有相邻节点。将节点 S 称为被泛洪数据分组的源节点。一个节点（比如 X）接收到该数据分组后，检索该分组头节点列表，并与自己的相邻节点列表比较（一个节点的相邻节点列表就是其所有相邻节点 ID 的列表）。假如其所有相邻节点在分组头节点列表中，那么 X 停止广播该数据分组。因此，避免了冗余数据传输。图 7-4 说明 MF 协议在一个所示配置中的操作。在通信链上标明分组头节点列表。在图 7-4 中，节点 A 需要对整个网络泛洪其感知数据，因此节点 A 给其所有相邻节点广播一个数据分组。节点 B 和 E 也广播该数据分组，但是该数据分组不会被进一步广播。



(a) MF 说明例子



(b) 虚拟栅格示例



(c) 一个虚拟栅格的网关节点和内部节点

图 7-4 LAF 协议描述示意图

尽管 MF 通过减少冗余传输能够节省能量，但是随着分组头节点列表变得越来越大，MF 的节能却越来越少。在经典泛洪中，每个节点严格广播一次，每个节点接收到其相邻节点的所有广播分组。因此，这个简单网络使用 6 个发送和 12 个接收来完成分组泛洪；而 MF 使用

4 个发送和 9 个接收来完成同一个分组泛洪。但是，假如由于分组头节点列表增大而导致分组长度增加 1 倍，那么 MF 使用 8 个发送和 18 个接收来泛洪一个分组。因此，分组头节点列表增大约束节能。下面介绍用位置辅助泛洪（LAF）来克服这种局限性。

2. 位置信息

LAF 使用位置信息将 WSN 划分成多个栅格。可以采用 7.1 节介绍的技术和其他定位技术获取传感器节点位置信息。下面的描述和讨论均假定每个传感器节点知道其精确位置。但是，在稍后的讨论中表明：LAF 能够容忍适度的位置估计误差以及相关大误差。

3. 虚拟栅格

LAF 将被监视区域（传感器场）划分成多个“虚拟栅格”。每个传感器节点根据其物理位置关联一个虚拟栅格，如图 7-4（b）所示，被监视区域被划分成 9 个虚拟栅格。节点 A 属于如图 7-4（b）所示的虚拟栅格。

4. LAF 节点类型

LAF 将传感器节点分成如下两种类型：

- ① 网关节点（Gateway Node）：假如节点 A 的任一相邻节点与 A 不属于同一个栅格，那么 A 就是网关节点。
- ② 内部节点（Internal Node）：假如节点 A 的所有相邻节点与 A 属于同一个栅格，那么 A 就是内部节点。

各个传感器节点布置完毕后运用其位置信息自动确定其虚拟栅格和状态（网关节点还是内部节点）。例如，在图 7-4（c）中，节点 A、G、F、I、D、H 为网关节点，而节点 B、C、E 为内部节点。网关节点将数据转发给其他栅格，内部节点在栅格内部转发数据。

5. 分组头格式

在 LAF 中使用的分组头格式如图 7-5 所示，其组成域包括：分组源节点身份识别码（SourceID）、分组序列号（SeqNumber）、接收节点列表（RecvNodeList）、栅格识别码（GridID）、节点类型（NodeType）。RecvNodeList 长度可变，列出已经接收到该分组的节点。GridID 表示该分组发送节点当前所在虚拟栅格的识别码。NodeType 说明该节点是网关节点还是内部节点。GridID 只由网关节点使用，用来防止对已经接收到分组的栅格重传该分组。每个组成域的长度最好由应用设计人员来确定。例如，某个 WSN 中的节点数量通常决定 GridID 的字节长度。

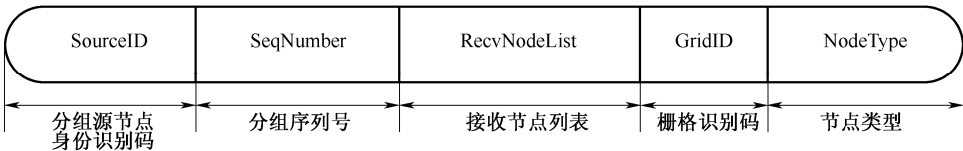


图 7-5 LAF 分组头格式

虚拟栅格的大小、虚拟栅格包含的节点数量依赖具体应用要求以及分组长度。假如分组短，那么经过几轮 MF 操作后，控制开销变得非常高。这里假定虚拟栅格数量是事先确定的。

7.3.2 采用LAF分发信息

1. 网关节点进行的数据转发

一个网关节点接收到其所在栅格内部的一个分组后，比较该分组头节点列表与自己的所有相邻节点，检查其所在栅格内是否存在还没有接收到该分组的相邻节点：若存在，则将这些相邻节点的 ID 添加到该分组头节点列表中，然后将该分组转发给这些还没有接收到该分组的相邻节点；若所有相邻节点已经接收到该分组，则不转发该分组。一个网关节点接收到另一个网关节点发送来的分组后，删掉该分组头节点列表中的内容，然后将自己的 ID、自己所有相邻节点的 ID 添加到分组头节点列表中，接着将分组转发给自己的所有相邻节点。因此，分组随着其传递通过不同的栅格而变短，但是随着在栅格内部的传递而变长。其 LAF 基本思想就是在分组中添加接收节点 ID，从而减少冗余传输。虚拟栅格用来限制分组的长度。LAF 网关节点存储最近所收分组的 SourceID 和 SeqNumber，从而使得网关节点能够防止分组在 WSN 中反复传输。

2. 内部节点进行的数据转发

一个内部节点接收到一个分组后，修改分组头节点列表，将其所有相邻节点的 ID 添加到该分组头节点列表中，然后将该分组转发给还没有接收到该分组的相邻节点。

LAF 是一个为无损耗网络设计的简单协议。但是，LAF 很容易适应有误差的通信链：将分组重传若干次，补偿有损耗通信链。LAF 不依赖节点的均匀布置，对于非均匀布置的 WSN，采用分布式方法很容易生成有规则的栅格，所以 LAF 也容易适应这种网络。

7.3.3 LAF中的资源管理

LAF 具有资源自适应能力。当各个传感器节点的剩余能量不相同时，剩余能量较少的节点可以选择首先等待一段超时时间，然后重传需要泛洪的分组。根据应用要求实现预置超时时间。LAF 关键思想就是剩余能量较少的节点应该只参与优先级高的应用任务，而优先级低的应用任务由剩余能量多的节点承担。这种方法允许所有传感器节点以相同程度参与所有任务，但是却会使剩余能量较少的节点较快耗尽其能量。这就能够更好地、更长时间地使用 WSN。

LAF 没有详细说明资源管理策略，而是将其交由应用来完成：根据应用的时延和网络寿命要求以及面向应用的其他要求，选择一个合适的资源管理策略。

7.3.4 栅格维护开销

传感器节点可以使用其虚拟栅格的任意独特属性作为 GridID。在稍后介绍的仿真实验中，传感器节点使用 (x,y) 作为 GridID， x 、 y 分别表示虚拟栅格左上角的 x 坐标、 y 坐标。对栅格

维护开销估计如下：假定分组头长 h 比特，组长 S 比特，被监视区域总共 n 个传感器节点且被划分成 N 个栅格，获取一个传感器节点位置信息所需要的信标消息为 n_B 条，接收每条信标消息所需能量为 E_B 。假如每个传感器节点计算栅格相关需要 n_p 个处理循环周期，那么计算栅格相关所需的总能量为 $n_p \times E_p$ ， E_p 表示一个处理循环周期所需要的能量。因此，维护栅格所需要的总能量为 $n \times [(n_B \times E_B) + (n_p \times E_p)]$ 。由此可见，栅格维护开销只与网络规模呈线性递增关系。

7.3.5 数据分发规程的完备性

下面证明 LAF 作为一个泛洪机制的完备性，即证明使用 LAF 总是能够完成数据泛洪。需要在网络中泛洪一个数据分组的一个节点成为该数据分组的源节点。证明：一个节点若是接收到采用经典泛洪的该源节点的数据分组，那么同样可以接收到采用 LAF 泛洪的这个数据分组。

引理 7-8：假如一个虚拟栅格上的一个网关节点接收到该数据分组，倘若该虚拟栅格上的所有节点通过经典泛洪接收到该数据分组，那么这些节点最终将会接收到这个数据分组。

证明：一个虚拟栅格上的每个节点或者为网关节点，或者为内部节点。考虑该虚拟栅格上的节点 A。将 A 的相邻区域表示为由 A 的所有相邻节点构成的集合 N_A 。考虑 N_A 中的节点 B 已经接收到该分组。假如 B 所收分组的分组头节点列表不包含节点 A，那么 B 将该分组转发给 A。但是根据 LAF 协议，只有当节点 A 收到该分组时，该分组头节点列表才会包含节点 A 的 ID。因此，节点 A 或者已经收到该分组，或者将从节点 B 接收该分组。节点 A 一旦接收到该分组，就会立即将其转发给还没有接收到该分组的所有相邻节点。所以，最终该虚拟栅格上的所有节点都将会接收到该分组。

定理 7-11：假如一个源节点给网络泛洪一条消息且每个节点采用 LAF 转发该消息，那么采用经典泛洪该消息能够传递到达网络中每个节点，则采用 LAF 泛洪该消息同样能够传递到达网络中每个节点。

证明：采用反证法证明。考虑随机 WSN 中一个节点采用经典泛洪接收到一条消息，但是采用 LAF 却没有接收到这条消息。将产生消息的节点称为源节点，而将正在考虑的节点称为目的节点，将源节点驻留的虚拟栅格称为源虚拟栅格，将目的节点驻留的虚拟栅格称为目的虚拟栅格。因为目的节点在经典泛洪中已经接收到该消息，所以存在一条从源节点到达目的节点的路径。根据引理 7-1，目的节点在 LAF 中没有接收到该消息意味着目的节点的虚拟栅格中没有网关节点接收到该消息。假如有任一网关节点接收到该消息，则该网关节点会将该消息转发给目的节点。这就意味着目的虚拟栅格没有任何相邻虚拟栅格接收到该消息。假如任一相邻虚拟栅格接收到该消息，那么该相邻虚拟栅格会将该消息转发给目的虚拟栅格的网关节点。按照类似方式继续进行推论，就能够证明源虚拟栅格的网关节点也接收不到该消息。这就意味着源虚拟栅格中没有泛洪消息，这显然是与事实矛盾的。因此，假如网络中每个节点执行 LAF 协议，那么每个节点最终都会接收到泛洪分组。

推论 7-3：假如只是在泛洪开始前发生节点失效，那么 LAF 的网络故障容忍程度等同于经典泛洪的网络故障容忍程度。

假定网络中一些节点在泛洪前失效。根据定理 7-1 知道：对于存在失效节点的 WSN，消息通过经典泛洪到达某个目的节点，那么消息通过 LAF 也同样能够到达该目的节点。因此，LAF 的故障容错能力等于经典泛洪的故障容错能力。

一个有趣的待解决问题是：假如在泛洪过程中发生节点失效，那么使用经典泛洪的容错能力补偿 LAF 的容错能力。

7.3.6 LAF节能分析

下面首先介绍两个简单拓扑并分析其 LAF 节能，并与经典泛洪的节能对比。然后，推导 LAF 在随机 WSN 中的节能公式。假定一条数据消息的平均长度为 S bit，网络直径为 D 个转发跳。一张图的直径等于图中任意两个节点间最短路径中的最大值。假如发送 1 bit 数据需要的能量为 E_T 、接收 1 bit 数据需要的能量为 E_R ，那么一个节点发送一条包含 k 个节点 ID 的数据消息，并且其一个相邻节点接收到该消息所消耗的能量为 $[S+(k \times i)] \times E_T + [S+(k \times i)] \times E_R$ ，其中 i 表示节点 ID 的比特长度。对于 N 个节点的全连通网络拓扑，对于需要泛洪的每个分组，需要 N 次发送、 $N(N-1)$ 次接收，因此全连通网络经典泛洪一个分组的总能耗 EC_{CF} 为

$$EC_{CF} = S \times N \times E_T + S \times N \times (N-1) \times E_R \quad (7-43)$$

在 LAF 中，由于发送的消息包含网络中所有节点的 ID，因此需要 1 次发送、 $(N-1)$ 次接收。假如忽略 LAF 分组长度的微小增大，那么全连通网络 LAF 泛洪一个分组的总能耗 EC_{CF} 为

$$EC_{LAF} = S \times E_T + S \times (N-1) \times E_R \quad (7-44)$$

假定 $N=30$ 、 $S=64$ B、 $E_T=0.8$ μ J/bit、 $E_R=0.6$ μ J/bit，那么全连通拓扑经典泛洪的能耗 $EC_{CF}=280$ mJ，而全连通拓扑 LAF 的能耗 $EC_{LAF}=9$ mJ。

考虑的第二个拓扑是 N 个节点的直线拓扑。每个节点最多两个相邻节点。对于需要泛洪的每个分组，需要 N 次发送、 $2(N-1)$ 次接收。这是因为在经典泛洪中，每个节点必须只对分组广播一次，因此需要 N 次发送；每个节点必须旁听其相邻节点的所有发送，因此总共需要 $2(N-1)$ 次接收。因此，直线网络拓扑经典泛洪一个分组的总能耗 EC_{CF} 为

$$EC_{CF} = S \times N \times E_T + 2 \times (N-1) \times S \times E_R \quad (7-45)$$

在 LAF 中，对于直线拓扑，一个分组每当被转发一次，消息长度则增大一次。对于其 ID 已经在分组头节点列表中的节点不会处理（即广播）该分组。因此，需要 $N-1$ 次发送、 N 次接收，第 N 个节点额外一次接收该分组，但是由于存在还没有接收到该分组的相邻节点而不会转发该分组。因此，直线拓扑网络 LAF 泛洪一个分组的总能耗 EC_{LAF} 为

$$\begin{aligned} EC_{LAF} &= ((1+2+\dots+(N-1)) \times i + (N-1) \times S) \times (E_T + E_R) \\ &= \{N \times [(N-1)/2] \times i + (n-1) \times S\} \times (E_T + E_R) + [(N-1) \times i + S] \times E_R \end{aligned} \quad (7-46)$$

仍然假定 $N=30$ 、 $S=64$ B、 $E_T=0.8$ μ J/bit、 $E_R=0.6$ μ J/bit、 $i=1$ B，那么直线拓扑经典泛洪的能耗 $EC_{CF}=12$ mJ，而直线拓扑 LAF 的能耗 $EC_{LAF}=6$ mJ。

下面分析在随机 WSN 中的节能，网络结构如下：传感器节点随机布置在一个矩形区域内。传感器节点依靠电池供电，传输距离有限。假如两个节点相互处在对方的传输距离 r 范围内，则这两个节点互为相邻节点。这类随机网络对于模拟涉及 Ad Hoc WSN 的大量实际情形很有用。现在推导 LAF 在这类网络中的节能预测公式。考虑一个随机网络总共 N 个节点，每个虚拟栅格 n 个节点。假定每个节点平均有 Δ 个相邻节点，平均 M 个相邻节点总是有分组需要发送。分组在传递过程中随着其分组头节点列中不断增加新节点 ID 而增大，但是这种分组增大相对于总分组长度很小而忽略不计。在 LAF 中，对于其 ID 已经在分组头节点列表中的节点不会处理（即广播）该分组。采用 MF 在一个虚拟栅格上泛洪一个分组的总能耗 E_V 为

$$E_V=(E_T+E_R \times M)(n) \times S \tag{7-47}$$

因此，在整个网络中泛洪一个分组的总能耗 EC_{LAF} 为

$$EC_{LAF}=(N/n) \times E_V \tag{7-48}$$

在经典泛洪中，泛洪一个分组的总能耗 EC_{CF} 为

$$EC_{CF}=(E_T+E_R \times \Delta) \times N \times S \tag{7-49}$$

7.3.7 位置估计中的误差

在上面的讨论和分析中，假定每个节点知道其精确的地理位置。但是，GPS 或者其他定位系统提供的位置估计可能存在误差。但是，位置不精确不会影响 LAF 的性能，理由有以下几点。①LAF 使用位置信息使一个节点关联一个特定虚拟栅格。假如位置估计误差导致节点假定相同虚拟栅格在一个不同的位置上，那么从数据分组角度来看这不会影响该节点的功能。②假如位置估计误差导致节点假定虚拟栅格的位置不同于虚拟栅格真正所在的位置，那么该节点就成为其所假定虚拟栅格的一个网关节点，这样仍然不会对 LAF 性能造成太大影响。类似地，大相关误差导致属于同一个虚拟栅格的一个节点组被错误地移动到不同的物理位置上，由于该节点组所有节点仍然属于同一个虚拟栅格，所以 LAF 性能未受影响。

7.3.8 LAF的性能

LAF 研究开发人员采用 C++开发了一个仿真器，用来评估 LAF 性能，并与其他数据分发算法进行比较。通过仿真实验发现：当数据分发时延相当时，LAF 的节能优于经典泛洪以及其他基于修剪的分发算法；采用 LAF 和 MF，节点密度较高（一跳相邻节点较多）的节点每单位能量分发的数据多于经典泛洪。因此，密集 WSN 为了节能，更适合采用 LAF 协议进行数据分发。

(1) 能量模型

假定每个传感器节点 20 m 传输距离，无线带宽 20 kb/s。表 7-2 给出传感器的典型特性（其数值来自 RF 单片集成电路 TR1000 电台的技术说明书）。

表 7-2 电台特性

电台工作方式	功耗/mW
发射 (Tx)	14.88
接收 (Rx)	12.50
空闲	12.36
休眠	0.016

(2) 仿真模型

开始采用 50 个节点的网络，网络位于 200 m×200 m 的监视区域内，见图 7-6 (f)。被监视区域分成 4 个虚拟栅格，平均 9 个网关节点。在全连通图形前提下随机生成该网络。发送一个分组的处理时延随机分布在 0~5 ms 之间，没有考虑排队时延和其他数据处理时延。对 LAF 数据分发协议仿真 200 次，求平均结果。每次运行时，随机选择一个节点向网络泛洪一个 64 B 分组。每个节点以 2 s 为周期广播一条 5 B 的 hello 消息。在仿真中没有实现定位系

统。为了较为准确地比较能耗,仿真每个节点以 2 s 为周期接收 3 条 10 字节信标消息。最后,假定网络无损耗。

尽管 LAF 依靠定位技术,但是为了简单,在仿真中没有考虑定位问题,而是利用仿真器提供的传感器节点地理位置来确定每个传感器节点的类型(在实际中,节点自动确定其状态)。但是,由于 LAF 的消息开销可忽略不计,所以这样处理位置问题不会对仿真结果产生明显影响。

(3) 系统获取的数据与时间的关系

图 7-6 (a) 表示经典泛洪、MF、LAF、自修剪^[40]和支配修剪五种数据分发协议的系统已分发数据与系统所耗时间之比率。如图 7-6 (a) 所示,这些分发协议间的消息时延之差可忽略不计。对于所有实际应用,可以忽略这些时延之差。时延之差随着 LAF 和修剪法的消息变长以及传播时间增大而增大。

(4) 系统能耗与时间的关系

图 7-6 (b) 给出了经典泛洪、MF、LAF、自修剪、支配修剪五种数据分发协议的系统能耗与系统分发时间之间的关系。如图 7-6 (b) 所示,LAF 的节能优势非常明显于泛洪协议。即使经过 35 ms 之后,LAF 的能耗仍然低于 20 mJ。其原因在于 LAF 通过使用少量状态信息减少了大量冗余传输。

(5) 栅格数量的影响

改变图 7-6 (f) 所示网络的虚拟栅格数量,通过仿真实验评估虚拟栅格数量对 LAF 性能的影响。图 7-6 (c) 给出了被监视区域分别划分成 1、4、8、50 个虚拟栅格的系统能耗。系统能耗随着虚拟栅格的增多而增加,增大到某个值后则下降。直观解释是:虚拟栅格较少时,分组中状态信息转发能耗通过分组长度的增大而得到补偿。当虚拟栅格较多时,分组长度保持在限制范围内,节能非常明显。但是,当虚拟栅格增加到每个虚拟栅格只有少量节点的时候,每个虚拟栅格内泛洪分组中承载的状态信息甚少,结果导致节能下降。

(6) 分组大小对节能的影响

WSN 中的分组长度典型值分别设为 32 B、64 B、96 B、128 B。泛洪分组增大,则节能提高。图 7-6 (d) 给出了分组长度分别为 64 B、96 B、128 B 的 LAF 节能(LAF 能耗与经典泛洪能耗之百分比)。

(7) 节点密度对节能的影响

图 7-6 (e) 表示平均节点密度对 LAF 节能的影响。100 个节点组成的网络被分成 8 个虚拟栅格,图 7-6 (e) 给出了将一个 64 B 数据分组分别分发给 90%、95%、99% 的节点的能耗与平均节点密度的变化关系曲线。通过改变传感器节点的位置来改变网络的平均节点密度。网络总能耗随着平均节点密度的提高而下降,其原因在于利用分组头节点列表中的信息避免了较多的冗余传输。

(8) 网络规模对 LAF 的影响

在 100~1 000 个节点范围内改变网络节点的数量,节点随机布置在一个 200 m×200 m 栅格内,整个栅格分成 8 个虚拟栅格,随机选择一个源节点泛洪一个分组,仿真研究 LAF 的网络可扩展性。实验结果说明:除了经典泛洪,所有其他数据分发协议(MF、自修剪、支配修剪)都是可扩展的,但是 LAF 的节能优于其他分发方法。

(9) 位置估计误差的影响

在上述仿真实验中人为引入节点位置估计误差,研究位置估计误差对 LAF 性能的影响。

在仿真中使每个传感器节点位置产生偏移，偏移范围是 $[x\pm e, y\pm e]$ ， e 表示位置估计误差（按照传感器节点传输距离的百分比来考虑偏移量）， $[x, y]$ 表示传感器节点的实际位置。节点使用这些人造错误位置信息建立起自己与虚拟栅格的关联关系，并确定自己的类型。通过仿真实验发现：高达 10% 的位置估计误差对 LAF 数据分组的能量效率、时延造成的影响可忽略不计。

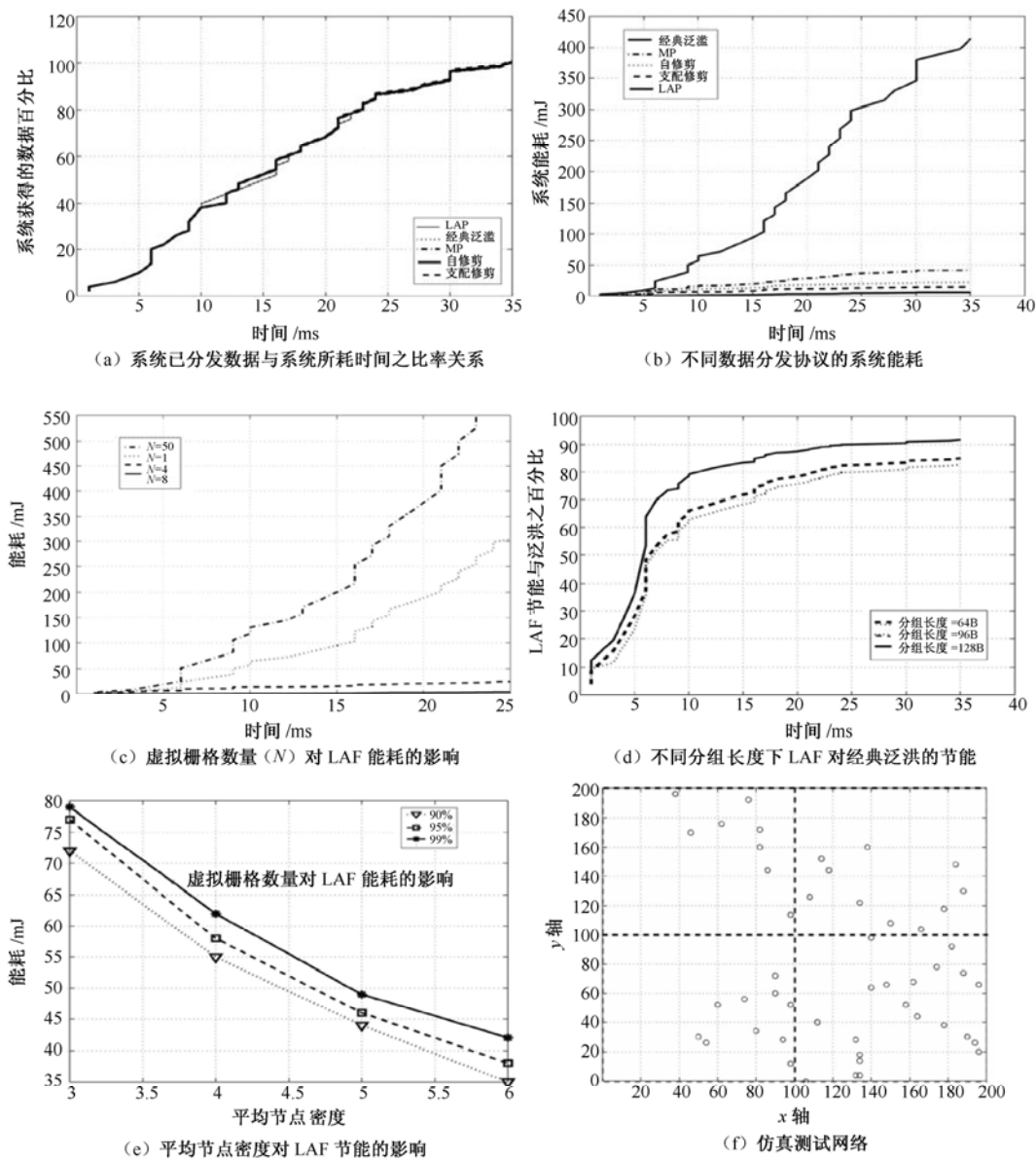


图 7-6 LAF 性能

参 考 文 献

[1] P. Bahl and V.N.Padmanabhan. RADAR: an In-Building RF-Based User Location and Tracking System. IEEE INFOCOM 2000, Vol.2, pp.775-784, 2000.

- [2] J.J. Caffery, Jr and G. L. Stiiber. Subscriber Location in CDMA Cellular Networks. IEEE Transactions on Vehicular Technology, Vol.47, No.2, pp.406-416, May 1998.
- [3] J.Hightower and G. Bomello. Location System for Ubiquitous Computing. IEEE Computer, Vol.34, No.8, pp.57-66, 2001.
- [4] B. Karp and H. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. MOBICOM, pp.243-254, 2000.
- [5] C. Savarese, J. Rabaey, and K. Langendoen. Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks. USENIX technical annual conference, Monterey, CA, pp.317-328, 2002.
- [6] A. Sawides, C.-C. Han and M.B. Srivastava. Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors. MOBICOM2001, pp.166-179, 2001.
- [7] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz. Localization From Mere Connectivity. MOBIHOC'03, pp.201-212, 2003.
- [8] Xiuzhen Cheng, Andrew Tbaeler, Guoliang Xue and Dechang Chen. TPS: A Time-Based Positioning Scheme for Outdoor Wireless Sensor Networks. IEEE INFOCOM 2004, pp.2685-2696, 2004.
- [9] F. Aurenhammer. Voronoi Diagrams—A Survey of a Fundamental Geometric Data Structure. ACM Computing Surveys, Vol.23, No.3, pp.345-405, 1991.
- [10] F. Baccelli, K. Tchoumatchenko, and S. Zuyev. Markov Paths on the Poisson-Delaunay Graph with Applications to Routing in Mobile Networks. Advances Applied Probability, vol.32, 2000.
- [11] P. Bose and P. Morin. Online Routing in Triangulations. ISAAC: Proc. 10th Int'l Symp. Algorithms and Computation, 1999.
- [12] K. Chakrabarty, S.S. Iyengar, H. Qi, and E. Cho. Grid Coverage for Surveillance and Target Location in Distributed Sensor Networks. IEEE Trans. Computers, Vol.51, No.12, pp.1448-1453, Dec.2002.
- [13] L. Chew. There Is a Planar Graph Almost as Good as the Complete Graph. Proc. Second Ann. ACM Symp. Computational Geometry, pp.169-177, 1986.
- [14] Crossbow, Mica, and Mica2 Wireless Measurement System Datasheets, 2003.
- [15] D.P. Dobkin, S.J. Friedman, and K.J. Supowit. Delaunay Graphs Are Almost as Good as Complete Graphs. Discrete and Computational Geometry, 1990.
- [16] M. Duarte and Y.-H. Hu. Distance Based Decision Fusion in a Distributed Wireless Sensor Network. Proc. Second Int'l Workshop Information Processing in Sensor Networks (IPSN 2003), Apr.2003.
- [17] D. Eppstein. Spanning Trees and Spanners. Technical Report ICS-TR-96-16, 1996.
- [18] J. Gao, L.J. Guibas, J. Hersherberger, L. Zhang, and A. Zhu. Geometric Spanner for Routing in Mobile Networks. Proc. Second ACM Symp. Mobile Ad Hoc Networking and Computing (MobiHoc '01), pp.45-55, Oct.2001.
- [19] G.L. Goodman. Detection and Classification for Unattended Ground Sensors. Proc. Information Decision and Control '99, pp.419-424, Feb.1999.

- [20] J.M. Keil and C.A. Gutwin. Classes of Graphs Which Approximate the Complete Euclidean Graph. *Discrete Computational Geometry*, no.7, 1992.
- [21] B. Karp and H.T. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. *Proc. Sixth ACM Int'l Conf. Mobile Computing and Networking (MobiCom '00)*, pp.243-254, 2000.
- [22] E. Kranakis, H. Singh, and J. Urrutia. Compass Routing on Geometric Networks. *Proc. 11th Canadian Conf. Computational Geometry*, pp.51-54, Aug.1999.
- [23] F. Kuhn, R. Wattenhofer, and A. Zollinger. Worst-Case Optimal and Average-Case Efficient Geometric Ad-Hoc Routing. *Proc. Fourth ACM Int'l Symp. Mobile Ad-Hoc Networking and Computing(MobiHoc)*, 2003.
- [24] K. Seada, M. Zuniga, A. Helmy, and B. Krishnamachari. Energy-Efficient Forwarding Strategies for Geographic Routing in Lossy Wireless Sensor Networks. *Proc. Second Int'l Conf. Embedded Networked Sensor Systems (SenSys'04)*, 2004.
- [25] Sensoria. SGate Datasheet. 2003.
- [26] D. Tian and N.D. Georganas. A Coverage-Preserved Node Scheduling Scheme for Large Wireless Sensor Networks. *Proc. First Int'l Workshop Wireless Sensor Networks and Applications(WSNA '02)*, pp.169-177, Sept.2002.
- [27] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C.D. Gill. Integrated Coverage and Connectivity Configuration for Energy Conservation in Sensor Networks. *ACM Trans. Sensor Networks*, Vol.1, No.1, 2005.
- [28] A. Woo, T. Tong, and D. Culler. Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks. *SenSys*, 2003.
- [29] G. Xing, C. Lu, R. Pless, and Q. Huang. On Greedy Geographic Routing Algorithms in Sensing-Covered Networks. *Proc. Fifth ACM Symp. Mobile Ad Hoc Networking and Computing (MobiHoc'04)*, pp.31-42, May 2004.
- [30] G. Xing, C. Lu, R. Pless, and Q. Huang. Impact of Sensing Coverage on Greedy Geographic Routing Algorithms. *IEEE Transactions on Parallel and Distributed Systems*, Vol.17, No.4, pp.348-360, April 2006.
- [31] T. Yan, T. He, and J.A. Stankovic. Differentiated Surveillance for Sensor Networks. *Proc. First Int'l Conf. Embedded Networked Sensor Systems (SenSys '03)*, 2003.
- [32] F. Ye, G. Zhong, S. Lu, and L. Zhang. PEAS: A Robust Energy Conserving Protocol for Long-Lived Sensor Networks. *Proc. 23rd Int'l Conf. Distributed Computing Systems (ICDCS '03)*, pp.169-177, May 2003.
- [33] H. Zhang and J.C. Hou. Maintaining Coverage and Connectivity in Large Sensor Networks. *The Wireless Ad Hoc and Sensor Networks: An Int'l J.*, 2005.
- [34] J. Zhao and R. Govindan. Understanding Packet Delivery Performance in Dense Wireless Sensor Networks. *Sensys*, Nov. 2003.
- [35] M. Zuniga and B. Krishnamachari. Analyzing the Transitional Region in Low Power Wireless Links. *Proc. First IEEE Int'l Conf. Sensor and Ad Hoc Comm. and Networks (SECON)*, Oct. 2004.

- [36] Harshavardhan Sabbineni and Krishnendu Chakrabartye. Location-Aided Flooding - An Energy-Efficient Data Dissemination Protocol for Wireless Sensor Networks. IEEE Transaction on Computers, Vol.54, No.1, pp.36-46, January 1997.
- [37] Ash Transceiver's Designers Guide. <http://www.rfm.com>, 2004.
- [38] R.R. Brooks, P. Ramanathan, and A.A. Sayeed. Distributed Target Classification and Tracking in Sensor Networks. Proc. IEEE, Vol.91, pp.1163-1171, Aug. 2003.
- [39] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An Energy-Efficient Co-Ordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks. Proc. ACM/IEEE Int'l Conf. Mobile Computing and Networking, pp.85-96, 2001.
- [40] H. Lim and C. Kim. Multicast Tree Construction and Flooding in Wireless Ad Hoc Networks. Proc. ACM Modeling, Analysis, and Simulation of Wireless and Mobile Systems, pp.61-68, 2000.
- [41] K. Whitehouse and D. Culler. Calibration as Parameter Estimation in Sensor Networks. Proc. ACM Int'l Workshop Sensor Networks and Applications, pp.59-67, 2002.
- [42] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An Energy-Efficient Co-Ordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks. Wireless Network 8, 481-494, 2002.
- [43] L. Feeney and M. Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. in: Proceedings of IEEE INFOCOM, Anchorage, AK (2001).
- [44] IEEE, Wireless LAN Medium Access Control and Physical Layer specifications, IEEE 802.11 Standard, IEEE Computer Society LAN MAN Standards Committee (August 1999).
- [45] B. Karp and H.T. Kung. GPSR: Greedy Perimeter Stateless Routing for wireless networks. in: Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), Boston, MA (August 2000).
- [46] J. Li, J. Jannotti, D.D. Couto, D. Karger and R. Morris. A scalable location service for geographic ad-hoc routing. in: Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom) (August 2000).
- [47] V. Rodoplu and T.H. Meng. Minimum energy mobile wireless networks. in: Proceedings of the IEEE International Conference on Communications (ICC), Vol. 3, Atlanta, GA (June 1998) pp.1633-1639.
- [48] C. Rohl, H. Woesner and A. Wolisz. A short look on power saving mechanisms in the wireless LAN standard draft IEEE 802.11. in: Proceedings of the the 6th WINLAB Workshop on Third Generation Wireless Systems, New Brunswick, NJ (March 1997).
- [49] T. Shepard. A channel access scheme for large dense packet radio networks. in: Proceedings of the ACM SIGCOMM, Stanford University. CA (August 1996) pp.219-230.
- [50] Y. Xu, J. Heidemann and D. Estrin, Geography-informed energy conservation for ad hoc routing. in: Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), Rome, Italy (July 2001) pp.70-84.

第 8 章 无线传感器网络端到端可靠传输协议

8.1 事件到中心节点的可靠传输协议 (ESRT)

WSN 是基于事件的系统, 系统依赖于许多微型传感器节点的共同努力。WSN 具有几个优点: 精确性更高、覆盖范围更广泛、本地提取法。要使这些变成现实, 所需要的事件特性必须可靠地传递给中心节点。除了需要性能优良的调制、媒介访问协议、链路差错控制、容错路由以外, 还需要可靠的传输机制。

由于相关数据流的绝对数量, 在 WSN 中不需要端到端可靠性, 中心节点必须以一定精度跟踪传感器场中的事件。因此在 WSN 中, 传输层上的事件到中心节点可靠性概念是必要概念。

事件到中心节点可靠传输协议 (Event-to-Sink Reliable Transport Protocol, ESRT) 以事件到中心节点可靠性为重点, 寻求以最低能耗和无拥塞方法实现可靠事件检测, 寻求满足 WSN 的独特要求。ESRT 协议适用于典型 WSN 应用, 包括事件检测、信号估计/跟踪。事件到中心节点可靠性概念使得 ESRT 协议不同于其他现有的、以端到端可靠性为重点的传输层模型。

通过对 ESRT 协议的扩充, ESRT 协议能够处理传感器场中的多个并发事件。传感器场中的并发事件并不是总是相互隔离的。由于各个事件流之间的交互作用, 所以不协调的协议操作可能得不到所需要的事件到中心节点的传输可靠性, 不能解决网络拥塞问题。因此, 有必要精确捕捉事件发生地点及采取相应的操作, 以最低能耗确保传感器场中所有并发事件到中心节点的可靠性。

8.1.1 问题定义

考虑一个典型 WSN 应用: 根据若干个传感器节点对所观测事件的联合报告可靠检测该事件和 (或者) 估计该事件特征。假定为了进行可靠的时间跟踪, 中心节点必须按周期 τ 个时间单位确定事件特征。因此, τ 表示一个决定间隔的持续时间, 是个固定值, 由应用层决定。在每个决定间隔结束之时, 中心节点根据在此期间从传感器节点接收到的报告作出决定。

假定中心节点在决定间隔 i 结束之时得到事件可靠性指示器 r_i 。必须只能采用中心节点的有效参数计算 r_i 。

按照所接收数据分组数量测量源节点事件特征的可靠传输。不管应用层实际采用的任何应用特定的参数, 所接收数据分组数量与中心节点检测和提取事件特征所需要的信息量密切相关。因此, 将接收数据分组数量作为传输层的一个简单而又精确的事件可靠性指标。所观察到的和所需要的事件可靠性定义如下:

定义 1: 观测事件可靠性 (Observed Event Reliability) r_i 等于中心节点在决定间隔 i 期间所接收到的数据分组数量。

定义 2: 所需事件可靠性 (Desired Event Reliability) R 等于事件可靠检测所需要的数据分组数量, 通常由应用决定。

假如观测事件可靠性 r_i 高于所需事件可靠性 R , 那么认为该事件可以被可靠检测。否则应该采取适当措施达到所需事件可靠性 R 。

假定传感器节点只要处在覆盖区域内并且观测到事件特征, 就立即将观测数据打包成分组, 然后将其发送给中心节点。由于每个传感器节点相对于事件中心的位置不同, 因而观测到的事件可能不同、发送给中心节点的分组也可能不同, 但是中心节点接收到的所有分组都被用来计算观察事件传输可靠性 r 。假定所有传感器感知数据的错误由传感器应用来处理, 但是使用中心节点所收数据分组来做出事件特征的实际决定。

采用上述定义, 给源节点数据分组添加事件 ID, 每当中心节点在决定间隔期间 i 中检测到一个事件 ID 时就将其接收分组计数器加一。这样做并不要求每个传感器节点有自己的 ID。将有关事件特征源信息量的递增模拟为传感器节点报告速率 f 的递增。

定义 3: 一个传感器节点的报告速率等于该节点单位时间发送的分组数量。

定义 4: WSN 的传输问题就是配置源节点报告速率 f 、在中心节点以最低资源利用率达到所需事件检测可靠性 R 的问题。

这种事件到中心节点可靠性概念的主要基本原理是: 传感器产生的数据在时间上相关, 在一定程度上容忍单个分组的丢失, 即中心节点估计事件特征的失真度 D_i 不会大于一定门限值 D_{\max} 。报告速率 f 跟采样速率、量化等级数量、感知特征数量等有关。因此, 报告速率 f 控制注入到传感器场中的流量大小, 同时调整物理现象相关采样值的数量。这反过来又会影响所观测事件的失真, 即事件检测可靠性。

实际上, 可以按照报告速率 f 的函数推导出中心节点在决定间隔 τ 期间的观测事件估计失真度 D 。假定一个事件信号 $s(t)$ 是一个高斯随机过程 $N(0, \sigma_s^2)$, 中心节点需要 $s(t)$ 在决定间隔 τ 上的期望值 $S(\tau)$ 。假定观测信号 $s(t)$ 是宽感知静态 (Wide-Sense Stationary, WSS) 信号且满足定义: $E\{S[n]\}=0$, $E\{(S[n])^2\}=\sigma_s^2$, $E\{S[n]S[m]\}=\sigma_s^2 \hat{\rho}_s(n, m)$, $E\{s(t)s(t+\delta)\}=\sigma_s^2 \rho_s(\delta)$, 其中协方差函数 $\hat{\rho}_s(n, m)=\rho_s\left(\frac{|m-n|}{f}\right)=e^{-\frac{|m-n|}{\theta}}$ 依赖信号样值间 (即 n 和 m) 的时间差、以及协方差系数 θ , 失真函数为^[5]

$$D(f)=\sigma_s^2+\frac{\sigma_s^4}{\tau f(\sigma_s^2+\sigma_N^2)}+\frac{\sigma_s^6}{\tau^2 f^2(\sigma_s^2+\sigma_N^2)^2}\sum_{k=1}^{\tau f}\sum_{l\neq k}e^{-\frac{\left(\frac{|k-l|}{f}\right)}{\theta}}-\frac{2\theta\sigma_s^4}{\tau^2 f^2(\sigma_s^2+\sigma_N^2)}\sum_{k=1}^{\tau f}\left(2-e^{-\frac{k}{f\theta}}-e^{-\frac{\left(\tau-\frac{k}{f}\right)}{\theta}}\right) \quad (8-1)$$

根据式 (8-1), 在被跟踪信号 S 的估计过程中观测到的失真 D 依赖传感器节点在决定间隔 i 期间用来向中心节点发送其感知数据的报告速率 f 。根据式 (8-1) 可绘出中心节点在报告速率 f 可变条件下的观察事件失真 D 的变化, 如图 8-1 (a) 所示。从式 (8-1) 和图 8-1 (a) 观察到: D 随着报告速率 f 的提高而下降。这是因为在决定间隔 i 期间所收采样数量随着 f 的提高而增大, 因此从事件区域传递给中心节点的信息量越大。报告速率 f 大于一定值之后, D 不会再随着 f 的提高而进一步下降。因此, 选择足够低的报告速率 f 而能达到一定的事件

失真范围（即所需事件可靠性目标 R ）能够节省大量能量，但是不会导致传感器稀缺资源的过度使用。这是 ESRT 协议的主要动机之一，以最低能耗实现事件的可靠传输。

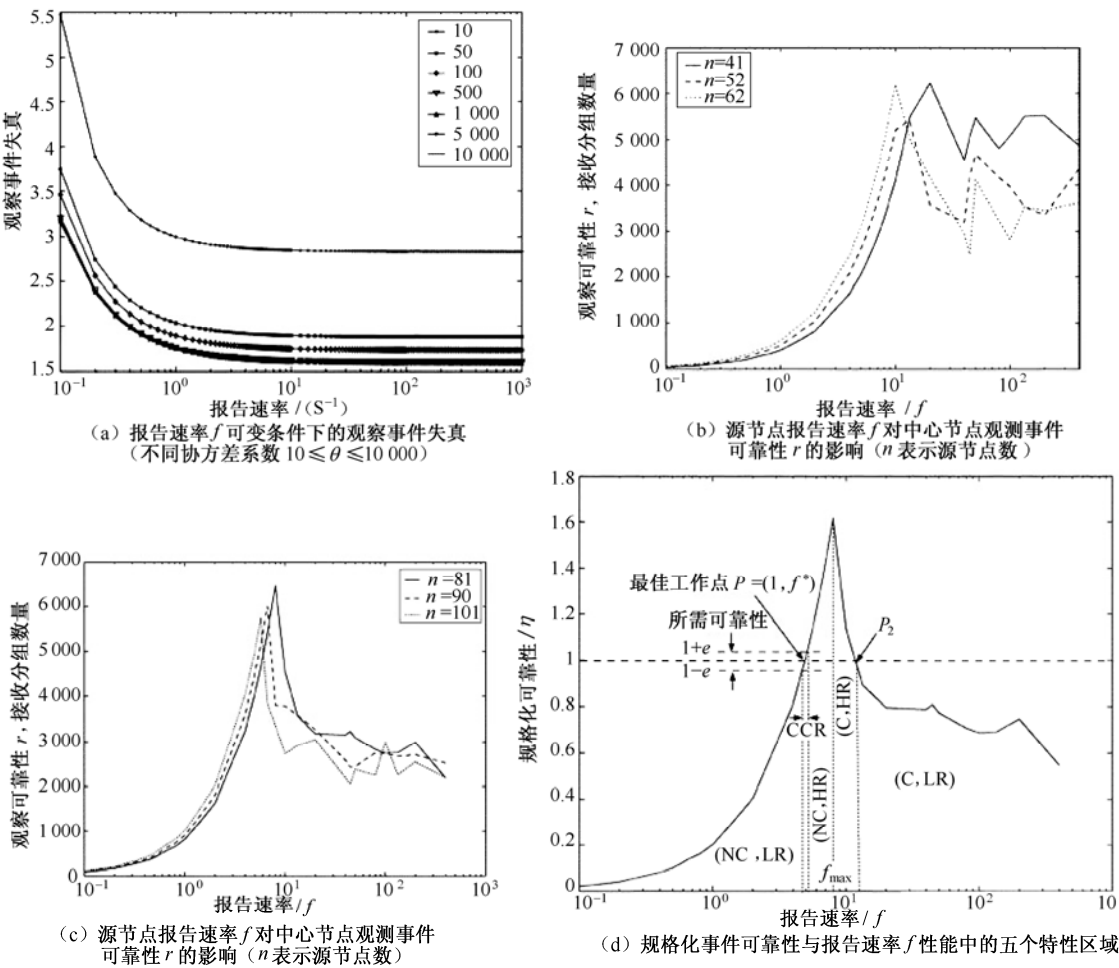


图 8-1 报告速率 f 的影响

任意选择低报告速率 f 来达到一定事件失真范围可能未必能够达到所需失真程度，因此未必能够确保事件传输的可靠性。这主要是因为：链路误码和网络连接中断导致按照选定的低报告速率 f 产生的所有采样可能不能全部被中心节点所接收到。另外，采用极高的报告速率 f 不会给观察事件失真带来有利影响，如图 8-1 (a) 所示，反而可能对事件传输的可靠性带来不利影响，这是因为极高报告速率 f 引起 WSN 拥塞。因此，必须有效控制报告速率 f ，以便事件特征的可靠传输，不会引起网络拥塞，能耗最低。这就是 ESRT 协议为了实现 WSN 的事件可靠传输而解决的主要问题。

8.1.2 评估环境

为了研究中心节点观测事件可靠性 r 和传感器节点报告速率 f 之间的关系，采用 ns-2 开发了一个评估环境，所用参数如表 8-1 所示。

表 8-1 ns-2 仿真参数

传 感 器 场	100 m×100 m
传感器节点数	200
传感器节点的无线传输距离	40 m
网络密度	每个节点 100 个相邻节点
分组长度	30 B
IFQ 长度	65 个分组
发射功率	0.660 W
接收功率	0.395 W
决定间隔 (τ)	10 s

200 个传感器节点随机分布在 100 m×100 m 的传感器场中，随机生成的网络拓扑固定不变。但是，传感器节点由于能量耗尽而失效，由此引起整个拓扑的变化。按照 Mica Mote 传感器的典型值仔细选择节点参数（比如传感器节点的无线传输距离）。从中选择一个传感器节点作为中心节点，所有源节点将其数据发送给这个中心节点。随机选择事件中心 (X_{ev}, Y_{ev})，事件范围内的所有传感器节点作为该事件的源节点。采用简单的 CSMA/CA MAC 协议和源动态路由协议 (DSR) 将源节点的感知数据发送给中心节点。采用其他路由协议对所实现的吞吐量的影响不明显。因此假定 r 与 f 的性能、ESRT 性能对低层路由协议不敏感是合理的。

图 8-1 (b) 给出了 41、52、62 个源节点的实验结果。通过改变事件中心 (X_{ev}, Y_{ev}) 的报告速率 f ，以及相应发送节点数量 n 得到图 8-1 (b) 中的每条曲线，表 8-2 列出了这些数据。对报告速率 f 的每个值做 5 次仿真实验，求事件可靠性实验结果的平均值 η 。时间范围固定在 30 m 范围内。

表 8-2 图 8-1 (b) 中 $n=41、52、62$ 三条曲线的事件中心

源节点数量	事件中心 (X_{ev}, Y_{ev})
41	(88.2, 62.8)
52	(32.6, 79.3)
62	(39.2, 58.1)

对图 8-1 (b) 作如下观察和分析：

- ① 当源节点报告速率 $f \leq f_{\max}$ 时，事件可靠性 r 与 f 呈线性递增关系；当 $f > f_{\max}$ 后，事件可靠性 r 下降。这是因为 WSN 由于拥塞而不能处理不断注入的数据分组，导致数据分组丢失。
- ② 事件可靠性 r 开始时递增而后来下降的变化现象与源节点数量 n 无关。
- ③ f_{\max} 随着源节点数量 n 的增大而减小，即源节点很多，但是源节点报告速率 f 较低时也会发生拥塞问题。
- ④ 当 $f > f_{\max}$ 后，事件可靠性 r 波动较大、不平稳。这种现象的直观解释是：接收分组数量（即事件可靠性 r ）等于源节点发送的数据分组总数 s 与网络丢失分组数量 d 之差。尽管 s 与 f 呈线性关系，但是 d 与 f 之间的关系不是线性关系。在有些情况下，即使网络已经拥塞， $s-d$ 却仍然递增。在最高事件可靠性 $f=f_{\max}$ 以下，事件可靠性 r 总是不存在波动性。

图 8-1 (c) 说明：进一步增大源节点数量 n ($n=81、90、101$)，事件可靠性 r 仍然存在

与图 8-1 (b) 类似的变化趋势。表 8-3 给出了图 8-1 (c) 的事件中心位置。在这组实验中，将事件范围固定在 40 m 范围内。

在图 8-1 (b) 中看到的 $f > f_{\max}$ 后事件可靠性 r 的波动现象在图 8-1 (c) 中仍然存在，但是图 8-1 (c) 中的 r 波动剧烈程度得到减弱，这是因为网络拥塞引起 r 下降得比较陡。从图 8-1 (c) 中可以看到图 8-1 (b) 中所有其他变化趋势。

从图 8-1 (c) 和图 8-1 (d) 中可以间接观察到实验中发送分组数量和成功接收分组数量。图 8-1 (c) 和图 8-1 (d) 按照事件覆盖范围内 n 个传感器节点采用报告速率 f 发送其感知数据时在决定间隔 τ 内接收到的数据分组数量来表示事件可靠性。在表 8-1 和图 8-1 (c)、图 8-1 (d) 中给出了 τ 、 n 、 f 的取值。因此，采用报告速率 f 在每个决定间隔 τ 内发送的分组数量可以计算为 $f \times \tau \times n$ 。发送分组数量与接收分组数量之比率等于 $(f \times \tau \times n) / r$ 。例如，在图 8-1 (d) 中，若 $f \approx 6.67$ 个分组/s， $\tau = 10$ s， $n = 101$ ，则接收分组数量 $r = 5\,746$ ，发送分组数量等于 $6.67 \times 10 \times 101 = 6\,736$ 。

采用密集布置进行评估，密集布置很可能发生拥塞。正如从图 8-1 (a) 和图 8-1 (b) 中看到的，随着发送数据分组的源节点的增多，网络能够承受的最大报告速率 f_{\max} 随着下降。但是 r - f 的一般表现却保持相同。因此，当网络密度没有密集布置时那么高时，在最大报告速率 f_{\max} 取较大值时发生拥塞。上面是直接针对 r - f 的一般表现进行讨论，得到的结果适用于网络密度较低情形。

表 8-3 图 8-1 (c) 中 $n=81$ 、90、101 三条曲线的事件中心

源节点数量	事件中心 (X_{ev}, Y_{ev})
81	(32.6, 79.3)
90	(61.1, 31.5)
101	(60.0, 63.6)

8.1.3 特性区域

现在仔细研究 r 与 f 的特征，确定五个特性区域，这五个特性区域对于 ESRT 协议的操作很重要。

考虑图 8-1 (c) 中 $n=81$ 个源节点的观测可靠性 r 曲线。为了方便，将这条曲线复制到图 8-1 (d) 中。下面全部采用这个特定情形来进行说明。但是已经验证：由于网络拥塞， r 与 f 的一般性能表现是开始时递增而后来下降， r 与 f 的这种表现趋势与参数取值无关。在图 8-1 (b) 和图 8-1 (c) 中均确实看到了 r 与 f 的这种表现趋势（改变 n 的取值）。因此，这里就针对特殊情形 ($n=81$) 的描述及其结果适用于具有任意参数数值集的 WSN 中的一般性 r 与 f 的性能表现。

设由应用决定的所需事件可靠性为 R ， $\eta = r/R$ 就是事件可靠性的测量。 η_i 表示在每个决定间隔 i 结束之时的规格化事件可靠性。

目标是尽可能接近 $\eta=1$ 工作，同时网络资源利用率最低[图 8-1 (d) 中的 f 逼近 f^*]。将此称为最佳工作点，图 8-1 (d) 中标有 P_1 。定义一个宽度等于 2ϵ 的容错范围，便于实际使用，如图 8-1 (d) 所示。 ϵ 是一个协议参数，稍后将详细描述。

直线 $\eta=1$ 与事件可靠性曲线相交于点 P_1 和 P_2 ，如图 8-1 (d) 所示。尽管在点 P_2 能够可靠检测事件，但是网络在点 P_2 已经拥塞，有些源节点发送的数据分组被丢失。只是由于源节

点报告速率高而弥补了网络拥塞造成的分组丢失，才得以实现事件可靠性，但是却造成有限储能的浪费，因此 P_2 不是最佳工作点。当 $\eta > 1 + \varepsilon$ 时，类似推理同样成立。

根据图 8-1 (d)，使用如下判决分界线确定五个特性区域（用虚线限制的区域）：

- (NC,LR)(No Congestion,Low Reliability): $f < f_{\max}$ 且 $\eta < 1 - \varepsilon$ (无拥塞，可靠性低)；
- (NC,HR)(No Congestion,High Reliability): $f \leq f_{\max}$ 且 $\eta > 1 + \varepsilon$ (无拥塞，可靠性高)；
- (C,HR)(Congestion,High Reliability): $f > f_{\max}$ 且 $\eta > 1$ (拥塞，可靠性高)；
- (C,LR)(Congestion,Low Reliability): $f > f_{\max}$ 且 $\eta \leq 1$ (拥塞，可靠性低)；
- OOR(Optimal Operating Region): $f < f_{\max}$ 且 $1 - \varepsilon \leq \eta \leq 1 + \varepsilon$ (最佳工作区)。

如前所述，中心节点在决定间隔 i 结束之时得到可靠性指示器 η_i 。联合拥塞检测机制（确定 $f > f_{\max}$ 还是 $f < f_{\max}$ ），有助于中心节点确定网络当前驻留在哪个特性区域中。因此，这五个特性区域确定网络的状态。设 S_i 表示在决定间隔 i 结束之时的网络状态。得到 $S_i \in \{(NC,LR), (NC,HR), (C,HR), (C,LR), OOR\}$ 。

ESRT 协议的操作与当前网络状态 S_i 密切相关。ESRT 协议的状态模型和状态转移如图 8-2 所示。

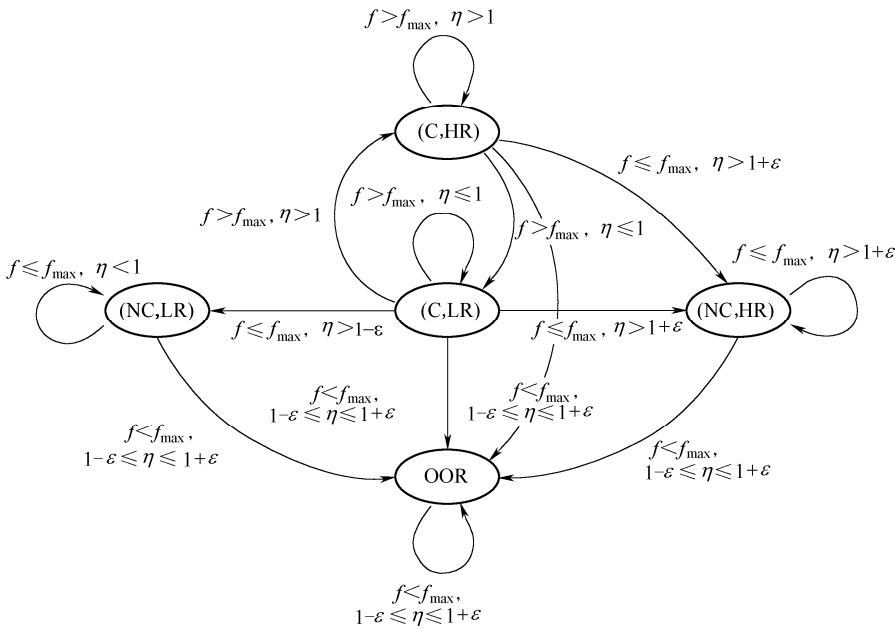


图 8-2 ESRT 协议的状态模型和状态转移

8.1.4 ESRT协议描述

ESRT 协议的主要目标是实现和维护网络在状态 OOR 下的操作。因此，目的是配置报告速率 f ，以最低能耗达到所需事件检测精度。为了有助于实现这个目的，ESRT 协议采用一个拥塞控制机制，该机制有两个方面的作用：一是事件的可靠检测，二是节能。

图 8-1 (d) 所示的 r 与 f 的变化特征随着传感器节点失效或者暂时关机造成的动态拓扑而变化。因此，高效传输协议应该连续跟踪中心节点观察到的可靠性，从而配置工作点。假

如 η_i 位于所需可靠性限制范围 ($1-\varepsilon \leq \eta_i \leq 1+\varepsilon$) 内, 并且没有接收到拥塞告警通知, 那么已经进入 OOR 状态, 中心节点通知源节点维持当前的报告速率 f_i 。这里假定中心节点有足够大的功率, 其广播发送可以到达所有源节点。

一般情况下, 网络可以驻留在任意一种状态中 $S_i \in \{(\text{NC}, \text{LR}), (\text{NC}, \text{HR}), (\text{C}, \text{HR}), (\text{C}, \text{LR}), \text{OOR}\}$ 。ESRT 协议依据当前状态 S_i 计算更新报告速率 f_{i+1} , 然后将 f_{i+1} 广播给所有源节点。例如, 假如 $S_i \in \{(\text{NC}, \text{LR}), (\text{C}, \text{LR})\}$, 那么观测事件可靠性等级对于所需事件特性的检测是不够的。在这种情况下, ESRT 协议主动更新报告速率, 以便尽可能地可靠跟踪事件。

ESRT 协议的自构特性有助于其适应动态拓扑和随机布置, 动态拓扑和随机布置在 WSN 中非常典型。ESRT 协议的另一个重要特性是其偏向于节省稀有资源, 同时可靠性等级又高于事件检测所需的可靠性。当 $S_i \in \{(\text{NC}, \text{HR}), (\text{C}, \text{HR})\}$ 时就是这样, 此时降低报告速率能够节省能量, 但是又不能降低事件检测的可靠性, 因此, ESRT 协议在这种情况下采用一种保守方法, 按照可控方式降低报告速率 f 。

ESRT 协议的算法主要由中心节点来运行, 而源节点只需要完成最低限度的功能, 即传感器节点只需要完成以下两个附加功能:

- ① 传感器节点必须在每个决定间隔结束之时侦听中心节点的广播, 更新其报告速率;
- ② 传感器节点必须使用简单、无开销、本地拥塞检测支撑机制。

第一个功能是实现问题, 稍后详细描述一个拥塞检测机制。将复杂性从源节点转移到中心节点, 既降低管理开销, 又节省宝贵的传感器资源。ESRT 协议利用中心节点将被更新的报告速率告知传感器节点, 既避免反馈时延, 又节省传感器节点能量资源。ESRT 协议按照联合识别原理工作, 不需要唯一的源节点 ID。

ESRT 协议利用前面定义的判决分界线并根据如下信息确定当前状态 S_i : ①中心节点在决定间隔 i 期间计算出来的可靠性指示器 η_i ; ②拥塞检测机制。

ESRT 协议根据当前状态 S_i 以及 η_i 和 f_i 的取值计算需要广播给源节点的更新报告速率 f_{i+1} 。中心节点在下一个决定间隔结束之时重新推导出一个与更新报告速率 f_{i+1} 对应的可靠性指示器 η_{i+1} 。为了结合任何拥塞报告, ESRT 协议重新确定网络状态 S_{i+1} 。反复重复这个过程, 直到进入最佳工作区 (状态 OOR) 为止。如图 8-2 所示, 并不是所有状态之间都能转移, 其原因在于 ESRT 协议采用的报告速率更新策略。针对每种网络状态的报告速率更新策略详细描述如下。

1. (NC,LR) 状态

在 (NC, LR) 状态, 不会出现拥塞, 所达到的可靠性低于所需可靠性, 即 $\eta < 1-\varepsilon$ 、 $f < f_{\max}$ 。这可能是如下一个或者几个原因造成的: ① 路由上的中间节点失效或者关电; ② 链路误码造成的分组丢失; ③ 源节点没有发送足够多的信息。

对于第①个原因, 需要通过这些节点传递的分组被丢掉。即使源节点发送足够多的信息, 由此也仍然可能导致可靠性下降。但是, WSN 定向扩散等路由算法提供容错路由或者重新建立路由。ESRT 协议可以与定向扩散等 WSN 路由协议一起工作。

在 WSN 中, 由于采用强误码纠错能力技术和重传技术的能量效率极低, 所以链路误码造成的分组丢失可能相当严重。但是, 不管分组错误率如何, 链路误码造成的分组丢失总数与报告速率 f 成正比。因此假定在连续若干个决定间隔内, 信道状况对分组丢失的影响效果不会出现明显偏差。在 WSN 应用中, 这个假设对于固定传感器节点、慢时变、空间隔离的

事件到中心节点的通信信道是合理的。因此，即使存在链路误码造成的分组丢失，开始时可靠性仍然是线性递增的。

为了将可靠性提高到可接收程度，需要增加源节点的信息。由于 ESRT 协议的主要目标是实现事件到中心节点的可靠性，所以 ESRT 协议主动增大报告速率 f ，尽可能获取所需可靠性。可以这样主动增大报告速率 f ：在无拥塞条件下，对于 $f < f_{\max}$ ， r 与 f 的关系是线性关系。因此可以使用如下乘法递增策略来计算报告速率更新 f_{i+1} ，即

$$f_{i+1} = f_i / \eta_i \tag{8-2}$$

式中， η_i 表示在决定间隔 i 结束之时中心节点观察到的可靠性。

2. (NC, HR) 状态

在 (NC, HR) 状态不会出现拥塞，所达到的可靠性高于所需可靠性，即 $\eta > 1 + \varepsilon$ 、 $f \leq f_{\max}$ 。这是因为源节点的报告频次高于所需的报告频次。因此，应该降低报告速率，以便节省能量。但是，降低报告速率时必须非常小心，以便总是能够维持事件到中心节点的可靠性。所以中心节点采用一种可控方式降低报告速率 f ：将斜率减半。直观上，这是在取消最大节能与失去可靠事件检测之间的平衡。因此，更新报告速率可以按如下公式计算：

$$f_{i+1} = (f_i/2) \times [1 + (1/\eta_i)] \tag{8-3}$$

这种更新策略既能够降低网络能耗，又不会给事件可靠性带来不利影响。

3. (C, HR) 状态

在 (C, HR) 状态，存在拥塞，所达到的可靠性高于所需可靠性，即 $\eta > 1 + \varepsilon$ 、 $f > f_{\max}$ 。这是因为 WSN 的独特特性，即使丢失一些源节点发送的数据分组，也仍然能够到达所需要的事件检测可靠性。在这种情形下，ESRT 协议可以降低报告速率、避免拥塞、节省传感器节点的能量。与 (NC, HR) 状态一样，降低报告速率时仍然必须小心，以便总是能够维持事件到中心节点的可靠性。但是，网络在 (C, HR) 状态下的工作比在 (NC, HR) 状态下的工作更远离最佳工作点。因此，需要采取更加主动的方法来减轻拥塞，并尽可能快地进入 (NC, HR) 状态。为此，使用式 (8-4) 乘法递减来仿效 (NC, HR) 状态的线性，采用乘法递减能够实现所有目标。

$$f_{i+1} = f_i / \eta_i \tag{8-4}$$

4. (C, LR) 状态

在 (C, LR) 状态，存在拥塞，观测可靠性不合格，即 $\eta \leq 1$ 、 $f > f_{\max}$ 。因为可靠性低、拥塞、能量被浪费，所以 (C, LR) 可能是最差状态。因此，ESRT 协议主动降低报告速率，使网络尽可能快地进入 OOR 状态。在 (C, LR) 状态下的可靠性不是报告速率的线性函数，如图 8-1 (d) 所示。为了确保充分降低报告速率，采用指数递减法，新的报告速率表示如下

$$f_{i+1} = f_i^{(\eta_i/k)} \tag{8-5}$$

式中， k 表示网络处在 (C, LR) 状态下的连续决定间隔个数，包括当前决定间隔，即 $k \geq 1$ 。假如没有检测到状态转移，则较主动地降低报告速率 f 。这个策略也能确保在 (C, LR) 状态下 $\eta=1$ 的收敛。

5. OOR状态

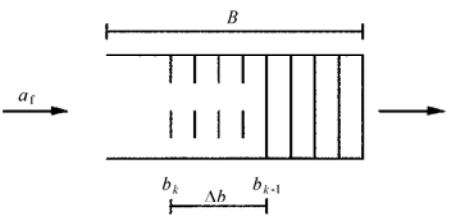
在 OOR 状态，网络在最佳工作点的 ε 误差范围内工作，在这个范围内以最低能耗得到所需可靠性。因此，下一个决定间隔的报告速率保持不变。图 8-3 (a) 按照伪码算法给出了完整的 ESRT 协议操作。

$$f_{i+1}=f_i$$

(8-6)

```
k=1
ESRT()
if(CONGESTION)
    if( $\eta < 1$ )
        /* State=(C, LR) */
        /* 主动降低报告速率 */
         $f = f^{\frac{2}{k}}$ 
        k=k+1;
    else if ( $\eta > 1$ )
        /* State=(C, HR) */
        /* 降低报告速率, 减轻拥塞,
        要求不影响可靠性。*/
        k=1;
         $f = \frac{f}{\eta}$ 
    end
else if (NO_CONGESTION)
    k=1;
    if ( $\eta < 1-\varepsilon$ )
        /* State=(NC, LR) */
        /* 提高报告速率 */
         $f = \frac{f}{\eta}$ 
    else if ( $\eta > 1+\varepsilon$ )
        /* State=(NC, HR) */
        /* 仔细降低报告速率 */
         $f = \frac{f}{2} \times (1 + \frac{1}{\eta})$ 
    else if ( $1-\varepsilon \leq \eta \leq 1+\varepsilon$ )
        /* 最佳工作区 */
        /* 报告速率保持不变 */
         $f = f$ 
    end
end
```

(a) ESRT 协议操作的算法



(b) 传感器节点缓存器等级监视示意图

事件 ID	CN (1 比特)	目的地	时戳	载荷	FEC
-------	--------------	-----	----	----	-----

(c) 包含拥塞通知域的典型数据分组结构

图 8-3 ESRT 的算法、缓存器等级监视、拥塞通知

8.1.5 拥塞检测

在 ESRT 协议中，为了确定当前网络状态 S_i ，中心节点必须具有网络拥塞检测能力。传统的基于 ACK/NACK 的检测方法用于端到端拥塞控制，对 WSN 不适用。原因还是在事件到中心节点可靠性概念上而不是在端到端可靠性概念上。只有中心节点能够确定可靠性指示器 η_i ，进行相应操作。端到端重传和 ACK/NACK 开销是对传感器节点有限资源的浪费。因此，ESRT 协议采用基于传感器节点本地缓存器等级监视的拥塞检测机制。任一传感器节点由于大量输入分组而导致其路由缓存器溢出，则是拥塞的，并将这个信息通知中心节点。这个拥塞检测机制详情描述如下。

在事件到中心节点的模型中，在每个报告周期 $1/f$ 内产生的流量主要依赖报告速率 f 和源节点数量 n 。由于报告速率 f 在每个决定间隔 $\tau > 1/f$ 结束之时受到中心节点的周期性控制，所以 f 在一个报告周期内不会发生变化。假定 n 在一个报告周期内不会发生重大变化，在下一个报告周期内产生的流量变化忽略不计。因此，假定任意传感器节点在连续若干个报告周期内的输入流量保持恒定。这反过来说明在每个报告周期结束之时缓存器填满程度的提高是恒定的。

设 b_k 和 b_{k-1} 分别表示第 k 个报告周期和第 $k-1$ 个报告周期结束之的缓存器填满程度， B 表示缓存器的大小，如图 8-3 (b) 所示。对于一个给定传感器节点，设 Δb 表示在最近一个报告周期结束之时观测到的缓存器长度增量，即

$$\Delta b = b_{k-1}-b_{k-1}$$

(8-7)

因此, 假如第 k 个报告周期结束之时的当前缓存器等级与最近的缓存器长度增量之和大于缓存器长度, 即 $b_{k-1} + \Delta b > B$, 那么这个传感器节点推断自己将在下一个报告周期内遇到拥塞问题。所以将其发送分组的分组头中的拥塞通知 CN (Congestion Notification) 位置位, 如图 8-3 (c) 所示 (CN 位置 1 表示向中心节点发出拥塞警示)。从而通知中心节点在下一个报告周期内将遇到上行拥塞情况。

假如中心节点接收到的分组的 CN 位被置位, 那么中心节点推断在最近一个决定间隔内遇到了拥塞问题。通过综合考虑可靠性指示器 η_i , 中心节点在决定间隔 i 结束之时确定当前网络状态 S_i , 并采取相应网络操作。

8.1.6 ESRT协议对并发事件的处理

可以将前面介绍的 ESRT 协议操作直接应用到发生单个事件的无线传感器场中。下面描述 ESRT 机制如何精确检测多个事件的发生、如何提取 ESRT 协议操作所需要的信息以及在多个事件并发条件下的 ESRT 协议操作。

1. 多个并发事件的检测

为了解决多个事件同时发生的问题, 有必要精确获取如下信息:

① 传感器场中是发生单个事件还是多个事件同时发生?

② 假如是多个事件同时发生, 那么产生的数据流从传感器节点往中心节点传递需要经过公共节点?

为了得到这两个问题的正确答案, 中心节点利用数据分组的事件 ID 域 (EventID) [见图 8-3 (c)]。EventID 域提供第①个问题的正确答案: 假如中心节点所收的所有数据分组包含一个相同的事件 ID (即所有数据分组的事件 ID 域的值全部相同), 那么无线传感器场中发生的是单个事件; 在这种情况下, 中心节点运用图 8-3 (a) 所示的 ESRT 协议操作算法达到所需的事件到中心节点的可靠性且能耗最低。假如中心节点所收数据分组包含不同的事件 ID (即各个数据分组的事件 ID 域的值不相同), 则中心节点推断传感器场中是多个事件同时发生。

注意: 已经隐含地假定采用任何现有高级网络信息收集机制 (比如网内数据累积法、数据累积位置意识路由、基于分群的事件识别法) 可以获得 EventID 或者分发 EventID。一个简单而可行的事件 ID 分配法就是动态随机事件 ID 分配策略, 每当第一次检测到一个事件的时候就初始化动态随机事件 ID 分配策略。于是, 第一个检测到事件的传感器节点随机选择一个 16 比特长的事件 ID。这个节点是第一个检测到该事件的节点, 产生携带有事件信息的数据分组, 捕捉无线通信信道, 将数据分组发送给中心节点。接收到这个本地广播的任何相邻节点均使用该分组中的事件 ID 作为其分组头的标记。因此, 这个随机选择的事件 ID 在事件覆盖区域内被动态传播。在事件覆盖区域的边界上停止事件 ID 的动态分发, 所以传感器转发节点对其转发的数据分组的 EventID 域不需要做任何修改。长度为 16 比特的事件 ID 的随机选择相当于一个 ID 冲突概率低于 10^{-5} , 这实际上是可以假定忽略不计的。另外, 当一个事件被一个传感器节点首次感知到的时候, 这个传感器节点为这个事件随机选择一个事件 ID, 并用这个事件 ID 广播自己的分组, 其间其他传感器节点也可能感知到这个事件, 并试图为这个事件分配 ID。但是, 由于感知到这个事件的第一个节点的本地广播而传输媒介忙,

所以其他传感器节点在 MAC 层推迟其广播，因而旁听到第一个节点的广播，然后在自己分组头的 EventID 域使用这个 ID。所以，对同一个事件产生两个不同的事件 ID 的可能性非常低。因此，这种动态随机事件 ID 分配策略不会引起 ID 冲突问题，可以安全使用。但是，多个事件并发条件下的 ESRT 协议操作不依赖特定的事件 ID 分配策略，所以很容易将其他 ID 分配方法综合到 ESRT 协议操作中。

假如传感器场中同时发生多个事件，则必须找到第②个问题的答案，即是否存在公共传感器节点作为这些事件流的传递路由器。这个信息不利于选择适当的 ESRT 协议操作，理由如下：假如不存在多个并发事件流传递的公共无线传感器节点，那么这些并发事件流是相互隔离的，即没有共享任何公共路径，如图 8-4（a）所示。因此，在这种情况下，ESRT 协议可以按照前面描述的默认操作分别单个处理各个并发事件的事件到中心节点的可靠性要求。

假如多个并发事件流需要通过一些公共无线传感器节点来传递，如图 8-4（b）所示，那么这些并发事件流不是孤立的。在这种情况下，分别单个处理每个事件流可能并不总是最佳解决方法。这是因为中心节点对这些并发事件流的任何操作都可能改变其可靠性等级和其他事件流的拥塞情况。因此协议操作必须慎重，而且还必须考虑无线传感器场中的所有并发事件流。

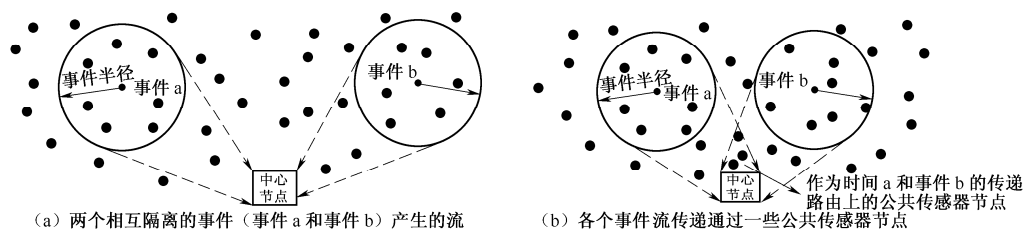


图 8-4 同一个无线传感器场中同时发生多个事件

为了确定必要的协议操作，中心节点必须精确检测多个事件流是否通过公共传感器节点。假如经检测确实存在公共传感器节点，那么必须知道哪几个事件流共享这些公共传感器节点。为此，中心节点利用数据分组的 EventID 域。假定数据分组的 EventID 域是多维域，能够包含多个并发事件的事件 ID。对执行路由的传感器节点的其他功能要求如下：

- ① 传感器节点维护一张事件列表（Event-list），在此表中记录事件的 ID，该传感器节点在无线传感器场中作为路由器节点。
- ② 路由器节点接收到一个新数据分组后，检查自己的事件列表以及该数据分组的多维 EventID 域，进行以下判断和处理：

假如事件列表中存在的一个事件 ID 不是该数据分组多维 EventID 域中的事件 ID，那么该传感器节点将其事件列表中的这个事件 ID 添加到该数据分组多维 EventID 域中的最前面，然后转发该数据分组。

假如事件列表中不存在这种事件 ID，那么该传感器节点检查其事件列表是否包含该数据分组多维 EventID 域中的第一个事件 ID：假如包含，那么该传感器节点对该数据分组头不作任何修改，直接转发该数据分组；假如不包含，那么该传感器节点将该数据分组多维 EventID 域中的第一个事件 ID 添加到自己的事件列表中，然后直接转发该数据分组（不作任何修改）。

为了说明精确检测多个并发事件的情况，假定一个传感器节点既转发事件 a 的数据分组

也转发事件 **b** 的数据分组，如图 8-4 (b) 所示。这个传感器节点知道自己作为路由器节点为事件 **a** 和事件 **b** 服务，因此其事件列表包含 **a** 和 **b**。现在假定这个传感器节点接收到一个其 EventID 域只包含 **c** 的数据分组，因此将 **a** 和 **b** 添加到这个数据分组的 EventID 域中，然后转发这个数据分组。由于这个传感器节点接收到事件 **c** 的数据分组，因此更新其事件列表（将 **c** 添加到该表中）。中心节点接收到这个数据分组（其 EventID 域包含 **c**、**a**、**b**）后，推断事件 **a**、**b**、**c** 的分组流不是相互隔离的，其间通过了一些公共传感器节点的传递，从而必须执行下面描述的多个并发事件下的 ESRT 协议操作。

2. 多个并发事件下的ESRT协议操作

根据上文的描述，中心节点利用数据分组的 EventID 域来捕获传感器场中发生多个事件的有关信息。

假如传感器场中发生单个事件，即中心节点接收到的所有数据分组承载同一个事件 ID，那么中心节点采用默认 ESRT 协议操作，使网络状态 *S* 进入最佳工作区 OOR。

对于多个并发事件，根据各个事件流是相互隔离的或者不是相互隔离的而选择不同的 ESRT 协议操作。下面详细描述这两种不同的 ESRT 协议操作。

(1) 相互隔离的多个并发事件

假如传感器场中存在多个并发事件，即中心节点接收到的数据分组具有不同的事件 ID，那么中心节点在决定间隔 *i* 结束之时检查所收数据分组 EventID 域。假如所有数据分组的多维 EventID 域只包含一个 ID，则中心节点推断各个事件流相互隔离，其传输路径不相交，没有公共传感器节点，如图 8-4 (a) 所示。

在这种情况下，设 S_i^k 和 f_i^k 分别表示事件 *k* 的当前网络状态和报告速率。ESRT 协议根据中心节点在决定间隔 *i* 期间计算出来的可靠性指示器 η_i^k 确定事件 *k* 的当前网络状态 S_i^k 。中心节点根据 S_i^k 、 η_i^k 、 f_i^k 计算更新报告速率 f_{i+1}^k ，并将 f_{i+1}^k 广播给事件 *k* 的事件范围内的传感器节点，以便将事件 *k* 的分组流的网络状态引入到最佳工作区 OOR。结果，中心节点采用默认 ESRT 协议操作分别达到各个事件的事件到中心节点的可靠性要求。

(2) 经过公共传感器节点传递的多个并发事件

假如中心节点所收数据分组 EventID 域包含不同的事件 ID，那么中心节点推断各个事件流传递经过若干个公共传感器节点，如图 8-4 (b) 所示。因此各个事件流不是相互隔离的。中心节点对其中任何一个事件流的操作都可能影响其他事件流的可靠性和拥塞情况。

在这种情况下，不能单独处理各个事件流，最好仔细考虑和处理无线传感器场中所有并发事件流。主要是因为 ESRT 协议的主要目标是实现事件到中心节点的可靠传输。处在不同网络状态下的事件流的 ESRT 协议操作的紧急程度不同。例如，在状态 (NC, HR) 下不存在网络拥塞，观测可靠性高于所需可靠性，但是在状态 (C, LR) 下既存在网络拥塞又没有达到事件到中心节点的可靠性，如图 8-1 (d) 所示。因此，当前状态为 (C, LR) 的事件流急需 ESRT 协议操作，中心节点对其操作的优先级较高。类似地，尽管在状态 (NC, LR) 和 (NC, HR) 下均不存在网络拥塞，但是当前状态为 (NC, LR) 的事件流没有达到所需可靠性，其网络操作的优先级高于状态 (NC, HR) 下的事件流。因此，根据每个网络状态的有关观测可靠性等级，将状态 {(C, LR), (NC, LR)} 归入高优先级状态，而将状态 {(C, HR), (NC, HR)} 归入低优先级状态。

中心节点根据共享公共传感器节点的各个并发事件的网络状态的优先级采取必要的操作。设 N_e 表示共享公共传感器节点的事件流的个数。从所收数据分组多维 EventID 域中获取这些事件的 ID。设 S_i^k 和 f_i^k 分别表示事件 k 的当前网络状态和报告速率, $k \in N_e$ 。

① 中心节点在决定间隔 i 结束之时确定每个事件 $k \in N_e$ 的分组流的网络状态 S_i^k 。

② 假如多个事件具有高优先级, 即 $\exists j \in N_e, S_i^j = (C, LR)$ 或者 $S_i^j = (NC, LR)$:

中心节点立即对这些事件执行默认 ESRT 协议操作, 即中心节点计算更新报告速率 f_{i+1}^j , 并将 f_{i+1}^j 广播给事件 j 的事件范围内的传感器节点, 即 $\forall j$ 有 $S_i^j = (C, LR)$ 或者 $S_i^j = (NC, LR)$ 。

比较紧急时才采取这个操作, 因为这些事件是不可靠传递给中心节点的, 因此第一个优先采取的操作就是使这些事件达到所需可靠性等级。

中心节点不更新其网络状态为低优先级的事件流的报告速率, 即 $f_{i+1}^j = f_i^j, \forall j$ 有 $S_i^j = (C, HR)$ 或者 $S_i^j = (NC, HR)$ 。

这是因为对高优先级网络状态采取的操作可能影响其可靠性已经较高的事件。因此, 避免了同时对这些事件流的进一步操作, 既达到最低能耗, 又没有影响可靠性等级。这与 ESRT 协议操作的主要目标是一致的。

③ 假如不存在高优先级网络状态的事件, 即 $S_i^j = (C, HR)$ 或者 $S_i^j = (NC, HR), \forall j \in N_e$, 那么中心节点对这些事件采用默认 ESRT 协议操作, 即计算更新报告速率 f_{i+1}^j , 并将 f_{i+1}^j 广播给事件 j 的事件范围内的传感器节点, $\forall j \in N_e$ 。

中心节点反复重复上述操作步骤, 直到所有事件流进入最佳工作区 OOR 为止。因此, ESRT 协议操作能够处理传感器场中的多个并发事件。

假如各个事件相互重叠, 即各个事件发生在相同附近区域内, 其有关事件覆盖区域相交, 那么 ESRT 协议采用正常操作按照单个统一事件处理这些相互重叠的事件。这些相互重叠事件的统一覆盖区域内的节点采用相同事件 ID, 而在统一覆盖区域边界上停止动态随机事件 ID 分发。

下面将介绍 ESRT 协议的分析结果和仿真结果。分析结果和仿真结果均证明 ESRT 协议从任一初始网络状态 $S_i \in \{(NC, LR), (NC, HR), (C, LR), (C, HR)\}$ 开始均能收敛到状态 OOR。

8.1.7 ESRT协议的性能分析

ESRT 协议的一些性能分析结果与初始网络状态 S_0 有关。

引理 8-1: 假如从状态 $S_0 = (NC, HR)$ 开始, 并且在网络不存在拥塞的时候具有线性可靠性 η , 那么 ESRT 协议收敛到状态 OOR 之前网络状态保持不变。

证明: $f < f_{\max}$ 时的线性可靠性 η 可以表示为 $f = \alpha\eta$, α 表示斜率。ESRT 协议保守递减 f 如下 [见式 (8-3)]:

$$f_{i+1} = (f_i/2) \times (1 + (1/\eta_i)) \quad (8-8)$$

因此,

$$\eta_{i+1} = (1 + \eta_i)/2 \quad (8-9)$$

因为根据式 (8-8) 有 $f_{i+1} < f_i$, 所以遵循 $S_i \in \{(NC, HR), (NC, LR), OOR\}, \forall i \geq 0$ 直到 ESRT 协议收敛。假如可能, 在 ESRT 协议收敛前, 当 $S_i = (NC, HR), j \geq 0$, 设 $S_{i+1} = (NC, HR)$ 。于是

$$\eta_{i+1} = (1 + \eta_i)/2 < 1 - \varepsilon \quad (8-10)$$

这就意味着 $\eta_i < 1 - 2\varepsilon$ ，但是由于 $S_i = (\text{NC}, \text{HR})$ ，而 $\eta_i > 1 + \varepsilon$ 。这里在 ESRT 协议收敛前有 $S_i \neq (\text{NC}, \text{LR})$ ， $i \geq 0$ 。综合前面的推论，在 ESRT 协议收敛到状态 OOR 前有 $S_i = (\text{NC}, \text{HR})$ ， $\forall i \geq 0$ 。

引理 8-2: 假如从状态 $S_0 = (\text{NC}, \text{HR})$ 开始，并且在网络不拥塞的时候具有线性可靠性 η ，那么 ESRT 在 $\tau \lceil \lg(\eta_0 - (1/\varepsilon)) \rceil$ 个时间单位内 ($\lg = \log_2$) 收敛到状态 OOR，其中 τ 表示决定间隔的长度。

证明: 设第 j 个决定间隔是 $S_j = \text{OOR}$ 时的第一个决定间隔。根据引理 8-1， j 是 $\eta_j < 1 + \varepsilon$ 成立的最小指数。利用式 (8-9)

$$\begin{aligned}\eta_j &= (\eta_{j-1} + 1)/2 < 1 + \varepsilon \\ \eta_{j-1} &= (\eta_{j-2} + 1)/2 < 1 + 2\varepsilon \\ &\dots \\ \eta_1 &= (\eta_0 + 1)/2 < 1 + (2^{j-1} \times \varepsilon)\end{aligned}\quad (8-11)$$

这里， $j > \lg[\eta_0 - (1/\varepsilon)]$ ，结果满足该条件。这就代表为了实现最大节能而到达 OOR 状态所需要的时间。通过采用保守递减式 (8-8) 方法，ESRT 协议可靠事件检测机制的主要目的始终得到维持。

引理 8-3: 关于网络不存在拥塞时具有线性可靠性 η ，对于任意 $i \geq 0$ ，网络状态转移 $S_i = (\text{C}, \text{HR}) \rightarrow S_{i+1} = (\text{NC}, \text{LR})$ 是不可能的。

证明: $f < f_{\max}$ 时的线性可靠性 η 可以表示为 $f = \alpha\eta$ ， α 表示斜率。从图 8-1 (b)、图 8-1 (c)、图 8-1 (d) 中 r 与 f 的关系特性中可以看到：对于状态 (C, HR) 下的每个 $f > f_{\max}$ ，存在一个 $f' < f_{\max}$ (在线性区域内) 使得 $\eta(f) = \eta(f')$ 成立。

现在采用反证法。假定当 $S_i = (\text{C}, \text{HR})$ 时 $S_{i+1} = (\text{NC}, \text{LR})$ ，对于有些 $i \geq 0$ 。根据前面的状态定义和更新策略，不等式

$$f'_i \times \frac{(1 - \varepsilon)}{\eta_i} > \frac{f_i}{\eta_i} \quad (8-12)$$

成立的必要条件是

$$f'_i > \frac{f_i}{1 - \varepsilon} > f_i \quad (8-13)$$

但是因为 $f_i > f_{\max} > f'_i$ ，所以这个必要条件不成立。这就完成了证明。根据这个结果，在图 8-2 所示的状态转移图中，不会实现从状态 (C, HR) 转移到状态 (NC, LR)。这就达到了减轻网络拥塞、降低能耗，同时不影响事件可靠性的目的。

为了确定 ESRT 协议从状态 $S_0 \in \{(\text{C}, \text{HR}), (\text{C}, \text{LR})\}$ 开始的收敛时间，需要跟踪分析 r 与 f 的非线性关系。下面通过仿真来证明这两种情况下的收敛。

8.1.8 ESRT协议的仿真结果

1. 单个事件的收敛时间

在 ns-2 中实现 ESRT，对 ESRT 进行仿真评估，研究 ESRT 协议的收敛问题。首先进行传感器场中发生单个事件的仿真实验。对每个仿真配置进行 5 次实验。使用表 8-1 所示的传感器节点和仿真配置。对于网络初始状态 $S_0 \in \{(\text{NC}, \text{LR}), (\text{NC}, \text{HR}), (\text{C}, \text{HR}), (\text{C}, \text{LR})\}$ 的收敛时

间实验结果如图 8-5 (a) ~图 8-5 (d) 所示。在图 8-5 (a) ~图 8-5 (d) 中，采用决定间隔数量和状态转移次数来表述 ESRT 协议的收敛图案，只是 f_i 和 η_i 稍有变化。因此，对每种网络初始状态 $S_0 \in \{(\text{NC}, \text{LR}), (\text{NC}, \text{HR}), (\text{C}, \text{HR}), (\text{C}, \text{LR})\}$ 分别给出一张收敛时间图案实验结果图，并且在每张图上列出 (f_i, η_i) 的取值以及状态。在所有仿真实验中，发送节点数 $n=81$ ，容忍公差 $\varepsilon=5\%$ 。事件覆盖半径固定为 40 m。其他仿真参数如表 8-1 所示。

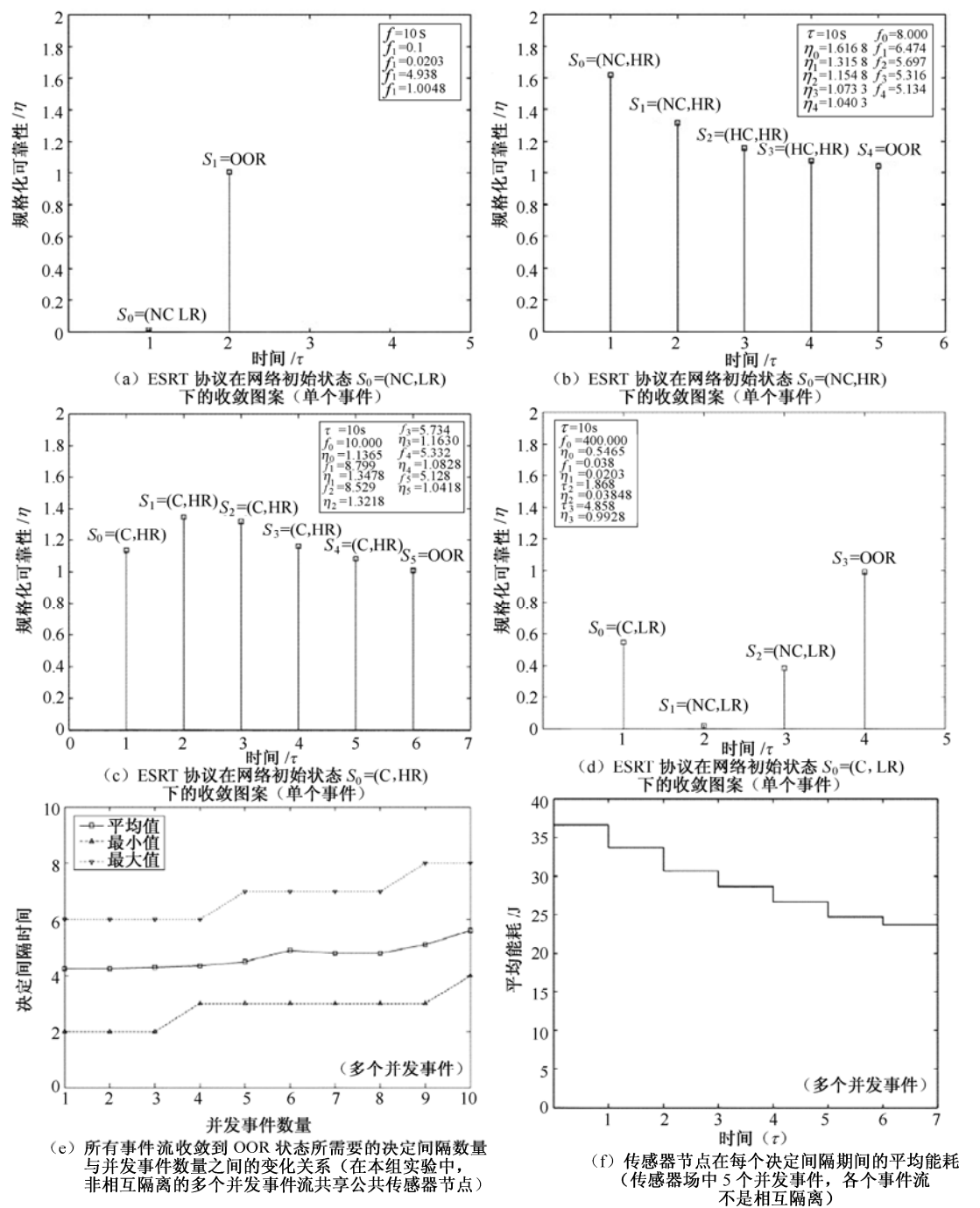


图 8-5 ESRT 仿真结果

从图 8-5 (a) ~图 8-5 (d) 中可以看到: ESRT 协议分别在网络初始状态 $S_0=(NC, LR)$ 、 (NC, HR) 、 (C, HR) 、 (C, LR) 下总共在 2 个 ($2\tau=20\text{ s}$)、5 个 ($5\tau=50\text{ s}$)、6 个 ($6\tau=60\text{ s}$)、4 个 ($4\tau=40\text{ s}$) 决定间隔时间内完成收敛。根据所采用的主动性乘法策略, 这是在预料之中的。根据 (f_i, η_i) 的取值以及图 8-5 (b) 和图 8-5 (c) 列出的状态可以验证引理 8-1、引理 8-2、引理 8-3。

2. 多个并发事件的收敛时间

对于 ESRT 协议在多个并发事件情况下的仿真实验, 同样采用表 8-1 所示的传感器节点和仿真配置。对每个仿真配置进行 5 次实验。事件发生在传感器场中随机位置上。图 8-5 (b) 分别给出了 5 次实验的平均结果。首先观察所有事件流收敛到 OOR 状态所需要的决定间隔数量及传感器节点的平均能耗。在仿真实验时, 改变并发事件的数量。

进行多个并发事件流不是相互隔离, 而是具有作为路由器的公共传感器节点进行仿真实验。如图 8-5 (e) 所示, 在这种条件下所有事件流收敛到 OOR 状态所需要的决定间隔数量随着并发事件的增多而稍有增多。主要是因为各个事件流不是相互隔离的, 因此 ESRT 协议需要考虑各个事件流当前网络状态的优先级。事件覆盖范围内的传感器节点已经具有足够高的可靠性, 因此不需要在每个决定间隔结束之时更新其报告速率。因此这些事件收敛到 OOR 状态所需要的决定间隔次数有所增多。由于同样的原因, 收敛所需的最小决定间隔个数和最大决定间隔个数也随着并发事件的增多而变化。从图 8-5 (e) 中看到, 在 10 个非相互隔离的并发事件下, 收敛时间的增加非常少。因此, ESRT 协议能够有效处理多个并发事件。

如图 8-5 (f) 所示, 在非相互隔离的多个并发事件下, 传感器节点的平均能耗表现与相互隔离的多个并发事件的平均能耗相同, 但是前者的平均能耗下降速度稍慢于后者。这是因为中心节点对已经具有足够高可靠性的事件流没有采取任何操作。这个结果与图 8-5 (e) 所示的平均收敛时间结果是一致的。

8.1.9 ε 的正确选择

在实际应用中, ESRT 协议在最佳工作点 P_1 [见图 8-1 (d)] 周围具有 ε 公差范围。假如在决定间隔 i 结束之时, 可靠性 η_i 在 $[1-\varepsilon, 1+\varepsilon]$ 内, 网络不存在拥塞, 那么网络处在 OOR 状态。中心节点认为事件是可靠检测的, 报告速率保持不变。因此, 利用 ε 小值就能够实现最佳工作点周围的较大邻区。但是, 根据引理 8-2, ε 越小, 收敛时间越长。所以 ε 的正确选择就是综合平衡公差要求和收敛要求的选择。例如, 当 $S_0=(NC, HR)$ 时, 1% 的公差要求能够抵消 7τ 时间单位的收敛时间, 但是根据引理 8-2, 由于保守递减, 所以仍然能够一直维持可靠的事件检测。

8.2 基于多电台虚拟中心节点的过载流量管理 (SIPHON)

实验 WSN (如 Mote WSN) 及其应用经常遇到周期性持续拥塞问题和高分组丢失率问题, 有时甚至发生拥塞崩溃问题。这会对真实中心节点的应用逼真度产生重大影响, 真实中心节点的应用逼真度是漏斗效应的直接结果, 多点到一点的多跳流量模式是 WSN 通信的特

点。因此必须考虑有关 WSN 流量过载管理问题。现有拥塞控制技术采用速率控制与分组丢失机制能够有效减轻拥塞问题，但是其代价是导致中心节点的应用逼真度明显降低。为了解决这个问题，建议采用少量全无线、多电台虚拟中心节点，这些虚拟中心节点可以随机分布在传感器场中，也可以选择性地布置在传感器场中。虚拟中心节点能够从传感器场的各个区域中提取数据事件。Siphon 是一组全分布式算法，支持 WSN 中的虚拟中心节点寻找与选择、拥塞检测、流量传输路径改道。Siphon 以虚拟中心节点星形网关实现为基础，虚拟中心节点采用独立的较长传输距离电台网络（以 IEEE 802.11 为基础）来提取事件，并将其发送给我一个或者多个真实中心节点，在提取点采用短距离 Mote 传感器电台来实现与传感器场的交互。分析结果、仿真结果、48 个 Mica2 Mote 传感器测试床实验结果表明：虚拟中心节点具有网络可扩展性，能够有效管理要求不断提高的流量，同时对应用逼真度的影响达到最小。

Siphon 算法包括：①虚拟中心节点（Virtual Sink，VS）寻找与可见度范围控制；②拥塞检测；③流量传输路径改道；④次网络的拥塞预防。其中拥塞检测采用了 CODA 机制，所以下面首先描述 CODA。

8.2.1 拥塞检测与预防（CODA）

1. CODA 的组成

拥塞检测与预防（COngestion Detection and Avoidance，CODA）是 WSN 的能量高效拥塞控制机制，由以下三个机制组成：

（1）拥塞检测机制

精确而高效的拥塞检测在无线网络拥塞控制中起着重要作用。CODA 综合运用当前与以往的信道载荷状态信息、当前缓存器缓存信息量来推断每个接收机的精确拥塞检测，开销低。因为 WSN 传输媒介是共享媒介，可能与相邻区域中其他装置发生流量拥塞，因此 WSN 必须知道信道状态。假如始终不停地侦听信道、测试本地载荷，则能量开销高。因此，CODA 采用采样技术，在适当时间监视本地信道，建立精确估计，能量开销最低。一旦检测到拥塞，则节点运用反压机制给其上行相邻节点发送信令。

（2）开环、逐跳反压

在 CODA 中，一个节点只要检测到拥塞就广播一条反压消息。反压消息按上行方向朝源节点传递。对于密集网络的脉冲数据事件，反压消息很可能直接传递给源节点。节点接收到反压消息后就可以根据本地拥塞策略（如分组丢失等）降低其发送速率或者丢掉分组。上行节点接收到反压消息后，根据其自己的本地网络状态确定是否还需要朝上行方向转发该反压消息。

（3）闭环、多源调整

在 CODA 中，闭环调整工作时间较慢，能够在发生持续性拥塞问题时针对单个中心节点发送事件分组的多个源节点插入拥塞控制。当源节点事件速率小于信道最大理论吞吐量的一定百分比的时候，源节点自行调整。但是，源节点事件速率大于这个门限值时，源节点很可能引起拥塞，因此触发闭环拥塞控制，源节点只进入中心节点调整。源节点需要中心节点恒定而慢速的反馈（如 ACK）来维护其事件速率。源节点接收到 ACK 作为时钟自调机制，允许源节点维护其当前事件速率。若源节点没有接收到 ACK，则迫使源节点降低其速率。

2. CSMA考虑

媒介访问控制对无线共享媒介中的数据脉冲管理性能起着极其重要的作用。人们不断努力设计适合于 WSN 的 TDMA 协议,节点周期性关电能够节省能量。TDMA 能够严格控制和安排网络中的流量传输,因而减轻了对拥塞控制的需求。但是,在 TDMA 广泛用于 WSN 之前,需要解决许多实际问题,包括同步、传输时间安排开销。

越来越多的 WSN 使用 CSMA 类协议作为 MAC 协议。例如,得到广泛应用的伯克利 Mote 传感器采用简单的 CSMA MAC 作为 TinyOS 平台的一个组成部分。

(1) 吞吐量

CODA 采用实践方法,假定 CSMA。CSMA 的理论最大吞吐量近似为^[8]:

$$S_{\max} \approx \frac{1}{(1+2\sqrt{\beta})} (\beta, 1) \quad (8-14)$$

式中, $\beta = \tau C/L$ 。 (8-15)

CSMA 的性能密切依赖 β 。 β 表示无线传播时延和信道空闲检测时延之和。 τ 表示时延,单位为秒, C 表示原始信道比特速率, L 表示一个数据分组的比特估计数。假如节点能够迅速检测到空闲周期,即 β 极小,则 CSMA 能够提供极高信道利用率,而与网络载荷无关。

式 (8-14) 给出了一跳范围内 CSMA 的信道容量。参考文献指出:一条理想 Ad Hoc 多跳转发链应该能够达到单跳传输所能达到的吞吐量的 25%。这个结果对于 CODA 设计具有重要指导意义。

(2) 隐含终端

CSMA 在多跳环境中会遇到隐含终端问题。IEEE 802.11 采用虚拟载波侦听(即 RTS/CTS 交互)来排除隐含终端。为了降低采用虚拟载波侦听而引入的信令开销,IEEE 802.11 传输短分组时不采用 RTS/CTS 交互。在 WSN 中,由于低占空因数要求和流量特点,分组通常较短(几十个字节),并且由于传感器节点的能量极有限,若传输每条消息都采用 RTS/CTS 交互则信令开销高、能耗高。

通常,在网络载荷很轻时,除了源节点和转发节点,其他节点大部分时间处于静默(关机)状态。因此,隐含终端引起的丢失分组极少。随机布置的节点在发送/转发分组时,即使在密集网络中,隐含终端的存在概率也非常低。

总之,就 WSN 而论,在正常操作期间,因开销高而没必要使用 VS。为了预防拥塞,需要设计一种能够满足使用或不使用 VS 的机制,要求在正常操作期间开销低,甚至开销等于零,足够快速解决拥塞问题。根据感知应用和无线技术,传输数据分组时不使用 VS,传输关键信令消息(比如路由协议控制分组)时采用 VS,以便降低开销。

(3) 链路层 ARQ

在 IEEE 802.11 MAC 中,将已发送的分组保存在发送缓存器中,直到接收到其 ACK 应答或者达到最大重传次数为止。这个机制提高了链路质量,代价是能耗和存储器空间的增大。但是,能量和存储器空间对传感器节点是稀缺资源,在正常操作下可能没必要一直用于支持链路可靠性(即由于 WSN 的应用特定性,并不是所有数据分组都需要严格的可靠性)。

在设计 WSN 协议时需要分别单独考虑可靠性和拥塞控制。对于关键信息交互(如路由信令),VC 和链路层 ARQ 作为通信可靠性手段是必需的;但是在拥塞期间,VC 和链路层

ARQ 就不是必需的。在 WSN 中，由于所分发数据中的固有冗余度，所以能耗比偶尔数据丢失更加重要。因此，主要目标功能是能耗最低化。这与 TCP 相反，丢失的 TCP 分组总是要恢复的。在设计 CODA 时，拥塞控制并不直接关心分组丢失，允许 CODA 消除与其他控制机制的可靠性之间的相互影响。因此，根据应用，CODA 能够与也可以不与可靠性机制一起使用。

3. 拥塞检测

CODA 综合运用缓存器队列长度、本地信道载荷、源节点报告速率/逼真度来检测拥塞。

(1) 缓存器队列长度

在传统数据网络中经常采用队列管理来检测拥塞。但是，由于没有链路层应答（有些应用可能不要求链路层应答，因此为了节能而不采用链路层应答），所以不能使用缓存器信息缓存量或者队列长度作为拥塞指示。为了进一步说明，利用 ns-2 仿真一个 5 节点的简单 IEEE 802.11 无线网络，如图 8-6（a）所示，节点 1 和 4 各自开始发送 CBR 分组，占 50%信道容量，CBR 分组通过节点 2 分别传递给节点 3 和 5。10 s 后，其中一个源节点停止发送。做两次仿真实验，一次实验使用 VS 和链路层 ARQ，另一次实验不使用 VS 和链路层 ARQ。

图 8-6（b）给出了时间序列轨迹结果，包括信道载荷、缓存器信息缓存量以及中间节点 2 的分组交付率。从图中看到：在两个源节点同时发送的时候，信道载荷几乎立即上升到 90%，发生拥塞，分组交付率从 100%下降到 20%左右；缓存器信息缓存量在拥塞期间（特别是在关闭 VS 仿真期间的时序序列轨迹中）缓慢增长，在仿真到 5 s 左右急剧下降（提供虚假的拥塞状态信息），这是因为没有链路层 ARQ，队列空并不意味着拥塞已减轻，因为出队列的分组可能由于碰撞而没有传递到达下一个转发跳。由于检测时延，CSMA 不保证相邻节点间无碰撞传输。

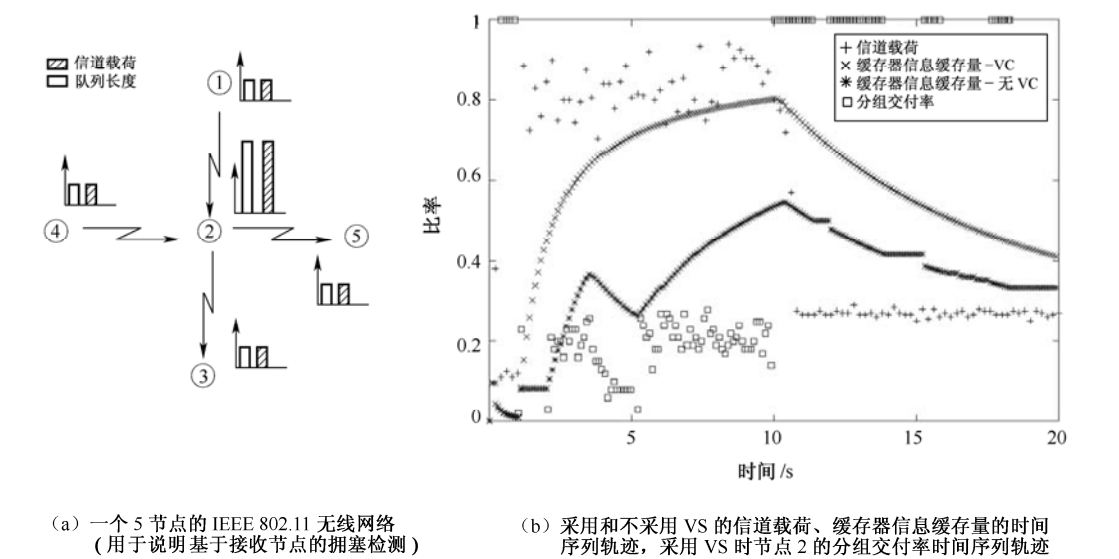


图 8-6 CODA 说明图

(2) 信道载荷

在 CSMA 网络中，传感器直接侦听信道，跟踪信道忙时间，计算本地信道载荷状态。

式 (8-14) 的 S_{\max} 给出了信道的最佳运用, 假如一个传感器检测到信道载荷达到信道容量的一定比例, 则表示发生碰撞的概率非常高。

信道载荷表示周围网络忙的精确程度, 是固有的本地拥塞减轻机制。例如, 信道载荷限制了由稀疏本地源节点产生的数据脉冲引起的大规模拥塞检测效果, 稀疏本地源节点产生高速率数据流量。信道侦听消耗节点相当大一部分能量。因此, 连续不停地侦听信道对 WSN 是不可取的。

(3) 报告速率/逼真度

对于典型的 WSN 应用, 中心节点要求一定的源节点采样速率或者报告速率。报告速率与应用密切相关, 可以作为事件逼真度指示, 即源节点关于某种现象的报告速率应该足够高, 以便满足应用要求的精确性。若中心节点的分组接收速率一直低于所要求的报告速率, 则推断传输路径上很可能是因为拥塞而在丢失分组。而在 ESRT 协议中, 除非中心节点接收到的分组已设置了拥塞通知比特, 否则若分组接收速率低于所要求的报告速率, 则 ESRT 协议给源节点发送信令, 要求提高源节点报告速率。

这种逼真度测试方法需要较长时间 (相对于分组传输时间), 并且考虑以下问题:

- ① 源节点和中心节点之间的端到端时延, 这是因为中心节点只知道自己的采样速率要求。
- ② 处理时延: 一个中心节点通常收集多个源节点关于同一个现象的数据 (如数据累积/融合)。为了处理不同源节点的数据分组沿着不同路径传递到达中心节点而带来的不同时延, 中心节点在作出结论之前需要等待一段最小时延, 以便收集各个源节点的报告。
- ③ 稳定性: 为了避免对瞬间现象 (可能引起抖动) 的不必要反应, 中心节点不应该对事件反应过快, 因此应该定义一个合适的较长时间的 “观察周期”。

总之, 基于报告速率的拥塞检测本来就慢, 而且是端到端的, 因此不能有效处理 WSN 中的短暂热点问题。

4. 开环逐跳反压

由于发生的拥塞各不相同, 所以传感器场中不同区域都可能发生热点问题。这推动了对 CODA 开环逐跳反压机制和闭环多源调整机制的需求。这两个机制互补性强, 而独立性不够。WSN 中不同节点需要不同的速率控制功能, 具体取决于节点是否为源节点、中心节点或中间节点。源节点知道流量特性, 中间节点不知道流量特性。按最佳方式布置中心节点, 以便于理解接收信号的逼真度率。在有些应用中, 中心节点是强能力节点, 能够进行复杂的直观推断。CODA 的目标是维护正常条件下的低开销或者零开销操作, 但相应速度必须足够快, 以便一旦检测到周围热点时迅速将其减轻或者排除拥塞。下面将详细描述 CODA 的反压机制和多源调整机制。

反压是在发生拥塞后采用的主要快速控制机制, 其主要思想是采用前面描述的各种检测技术 (队列长度、信道载荷、报告速率) 进行本地各个节点的低开销拥塞检测。一旦检测到拥塞, 接收节点给其相邻节点广播一条抑制消息, 同时作出本地调整, 防止拥塞向下行方向扩散。

节点只要检测到拥塞, 就立即广播反压消息。反压消息朝源节点往上行方向传递。对于密集网络的脉冲数据事件, 反压消息很可能直接传递给源节点。节点接收到反压消息后就可以根据某些本地拥塞策略 (比如丢掉分组、AIMD 等) 减慢其发送速率或者将分组丢掉。

上行节点接收到反压消息后, 根据其本地网络状态决定是否需要朝上行方向进一步传播

反压消息。例如，节点接收到反压消息后，根据本地拥塞策略可以将其输入数据分组丢掉，防止其队列填满而堵塞，这是由于队列溢出而不会朝上行方向进一步传播反压消息。但是，由于节点就地处理拥塞指示，因此需要闭环拥塞控制机制来处理这种情况下的持续性拥塞，不会传播反压消息。

采用拥塞深度(Depth of Congestion)来表示反压消息在遭遇拥塞前传递通过的转发跳数。路由协议和本地分组丢失策略可以采用拥塞深度来衡量通过不同路径的能耗。可以采用如下两个简单方案：①瞬间拥塞深度作为指示，路由协议据此选择较佳路径，从而减轻遭遇深度拥塞的路径上的流量。②节点可以抑制或者丢掉有关路由协议或者数据分发协议的重要信令消息（比如定向扩散协议中的兴趣、SPIN 数据广播等）。这种方法有助于事件流按照透明方式离开拥塞区域和远离热点。

（1）基于接收方的检测

以下两种情况适合拥塞指示：

① 快要溢出的队列。

② 所测信道载荷高于最佳信道利用率的一定比例。这是拥塞的概率指示，通过观测信道载荷接近上限值的程度来获得。

队列长度监视除了极低的处理开销外几乎没有开销，只提供双峰指示。信道侦听要么测量信道载荷、要么获取碰撞检测信令信息，提供合适拥塞指示，但若是持续不停地侦听信道则能量开销高。因此只在合适时候激活信道侦听，使能量开销最低是非常关键的。

考虑一个 WSN 节点的典型分组转发及其正常电台工作方式。电台只要不在发送状态或者已经关机，则处在侦听方式。当检测到信道上的载波后，电台切换到接收方式，等待发送前导，然后继续接收分组比特流。在将分组转发给下一个转发跳之前，CSMA 要求电台检测空闲信道，即电台对信道侦听一段时间。假如信道在侦听期间空闲，那么电台切换到发送方式，发送一个分组。由于需要在分组发送前进行载波侦听，所以当节点需要发送一个分组时，不存在额外的信道载荷侦听和测试开销。因此，激活检测机制的合适时间就是节点发送缓存器不空的时候，即根据某些协调机制（比如 SPAN、S-MAC 等）节点电台可能大部分时间处于关闭状态，但是，只要需要接收或者发送分组，电台至少必须处在侦听方式。

图 8-6 (a) 表示 WSN 中一种可能产生热点或者拥塞区域的典型情况：节点 1 和 4 发送 CBR 流量，消耗一半信道容量，CBR 流量经过节点 2 分别到达节点 3 和 5。由于信道太忙，所以节点 2 将其所收分组存储在自己队列中，最终丢掉所缓存的分组。这个简单例子说明：在拥塞相邻区域内，接收节点[如图 8-6(a)中的节点 2]的缓存器缓存信息量高或者至少不空。节点在缓存器不空时激活信道载荷测试响应率高，几乎没有开销。当缓存器变空时，信道载荷测试自动停止，这说明拥塞得到减轻、数据流平稳通过相邻区域周围的概率高。因此，假如节点只在其接收分组并需要转发该分组之时激活信道载荷监视，那么信道载荷测试额外开销甚低。CODA 需要进行信道载荷测试的唯一时间就是节点需要发送的时候，并且必须在发送之前进行载波侦听。

（2）最低开销采样

侦听周期定义为分组时间的整数倍。当一个节点开始侦听信道（即有信息需要发送）的时候，要求 MAC 协议至少对信道侦听一个侦听周期，以便测试信道载荷。在一个侦听周期期间，节点不是连续不停地在退避期间侦听，而是进行周期性采样，以便节省能量，在采样之间的间隔期间可以关掉电台。在测试 N 个连续长度为 E 的侦听周期的信道载荷时运用采样

方案，采用预先定义的采样速率获取信道状态信息，即在一个侦听周期内信道状态忙与空闲的时间倍数。然后按照 Φ_n （在侦听周期 n 期间的信道载荷测试结果）的指数平均以及参数 α 计算前 N 个连续侦听周期的信道载荷侦听结果 $\bar{\Phi}$ ，如式（8-16）所示

$$\bar{\Phi}_{n+1} = \alpha \bar{\Phi} + (1 - \alpha) \bar{\Phi}_n, \quad n \in \{1, 2, \dots, N\}, \bar{\Phi}_1 = \Phi_1 \quad (8-16)$$

假如发送缓存器在 n 到达 N 之前变空，那么忽略平均值，将 n 设为 1。数组 (N, E, α) 提供一种调整采样方案的方法，用于对特定无线系统体系结构的信道载荷精确测试。

（3）消息抑制

当所测信道载荷大于门限值（ S_{\max} ）时，则意味着发生了拥塞。节点广播抑制消息（作为反压消息），同时执行本地拥塞策略。尽管不能保证所有相邻节点都能接收到抑制消息，但是至少有一些节点能够以一定概率接收到抑制消息。节点根据信道载荷和缓存器信息缓存量检测到拥塞后就广播抑制消息。只要拥塞持续存在，节点就继续广播抑制消息，直到达到预定最大广播次数为止，相邻两次广播之间的时间间隔最短。

抑制消息是开环反压机器的基础，还可以作为按需 CTS 信令，所有其他相邻节点（发送节点除外，随机选择的发送节点，或者是能够分配较多机会给较迫切需要的发送节点的节点）至少可以静默一个分组传输时间，从而支持 CODA 隐含的优先级方法，即可以根据数据类型或者其他参数选择嵌入了抑制消息的“已选节点”，必须给已选发送节点分配较高的信道使用优先级。所有节点共享数据类型的优先级列表，一种确定数据类型具有高于其他数据类型的优先级。

5. 闭环多源调整

在 WSN 中，如果发生持续性拥塞，则需要维护多个源节点对单个中心节点的拥塞控制，中心节点作为 1 到 N 控制器对多个源节点起着重要的作用。闭环流控需要反馈信令，因此其开销高（相对于简单的开环流控）。下面介绍一种流控方法，用于动态调整跟特定数据事件有关的所有源节点。在正常操作下，各个源节点按照预先确定的速率自行调整（比如根据定向扩散或者 SPIN 数据分发协议），不会干预闭环中心节点调整。

当源节点事件速率 r 小于信道最大理论吞吐量 S_{\max} 一定比例 η 时，源节点自行调整；当源节点事件速率 r 大于该值（ $r \geq \eta S_{\max}$ ）时，源节点很可能引起拥塞，因此触发闭环控制。这个门限 η 不同于本地拥塞检测使用的门限，实际上前者比后者小得多。假如大于这个门限 η ，那么源节点只进入中心节点调整。此时，源节点不断需要中心节点的反馈信息（比如 ACK）来维持其事件速率 r 。源节点检测到（ $r \geq \eta S_{\max}$ ）后触发中心节点调整，在其转发给中心节点的事件分组中设置调整比特。接收到调整比特被置位的分组后，迫使中心节点回送 ACK（比如中心节点每接收到 100 个事件回送一个 ACK），以便调整跟特定数据事件有关的所有源节点。按照应用特定方式发送 ACK。例如，中心节点只沿着定向扩散应用需要强化的路径发送 ACK。源节点接收到 ACK 可以作为自行计时机制，以便允许源节点维护当前事件速率 r 。

源节点设置调整比特，希望按预定速率接收到中心节点回送的 ACK 或者在预定周期内接收到中心节点回送的一定数量 ACK，容忍短暂拥塞造成的 ACK 偶尔丢失。源节点在预定周期内接收到预定数量 ACK，则维持其事件速率 r 。当发生拥塞时，ACK 被丢失，迫使源节点按照某种速率递减函数（比如乘法递减等）下调其事件速率。中心节点也可能根据所观测

到的网络状态停止发送 ACK。中心节点能够测试其本地信道载荷 ρ ，假如 $\rho \geq \gamma s_{\max}$ ，则停止给源节点回送 ACK。

由于中心节点要求一定的事件报告速率，所以当事件报告速率一直低于所需报告速率（即信号逼真度）时，中心节点也可以采用应用特定操作。当事件报告速率一直低于所需报告速率时，中心节点推断路径上的事件分组由于持续性拥塞而被丢失，并且停止给源节点发送 ACK。当拥塞被排除后，中心节点又可以重新开始发送 ACK，因此，源节点按照某种速率递增函数（比如加法递增等）上调其事件速率。

由于有些应用的中心节点能力强于传感器，中心节点是数据收集点，所以中心节点能够维护有关特定数据类型状态信息。通过观测源节点的分组流，假如推断发生了拥塞，那么中心节点可以直接给源节点发送控制消息，以便降低源节点门限值 η ，迫使源节点以较低速率（即低于较重要的观测中心节点）触发中心节点调整。从而提供一个间接优先权机制，作为闭环拥塞控制的一个组成部分。

当源节点事件速率 r 被复位（比如定向扩散协议中通过强化）而使得 $r < \eta s_{\max}$ 时，源节点又开始自行调整，但是不需要中心节点回送的 ACK。这种多模拥塞控制技术为设计高效、低开销、可以在伯克利 Mote WSN 上实现的拥塞控制技术提供了基础。总之，相对于开环拥塞控制，闭环多源调整与应用层的关系更加密切。

ESRT 总是采用大功率以及一个公共速率直接给所有源节点发送一条调整消息，实现对所有源节点的调整。CODA 只调整其数据事件造成或者加重拥塞，或者其数据事件被传输路径上热点所阻止的有关源节点。当源节点请求调整时，CODA 没有采用高功率发射的单条控制消息，而是采用逐跳信令传递法来实现调整。

8.2.2 虚拟中心节点寻找与可见度范围控制

SIPHON 用于栖息地、流量处理。特别是真实中心节点可能没有配备第二套电台，因此，不能保证虚拟中心节点通过其远距离电台构成以真实中心节点为树根的次连通网络。而且，由于对 VS 布局要求相对较稀疏，所以不能保证一个 VS 与一个拥塞区域相邻。因此，拥塞节点需要寻找本地 VS 的能量高效方法，本地 VS 可能相距数个转发跳远。

SIPHON 采用带内信令法，将信令字节嵌入到真实中心节点周期性产生和发送的控制分组中。在典型的 WSN 应用中，要求真实中心节点周期性对网络发送信令，实现对网络的管理。例如，定向扩散协议要求周期性刷新兴趣；TinyOS 中的多跳路由（MultiHopRouter）协议（用于 Mica Mote WSN）要求网络中每个节点周期性广播路由控制消息，用于估计路由开销和监视链路质量。在这些情况下，可以自由嵌入 SIPHON 的信令字节，实现近似零开销的 VS 寻找。对于不需要中心节点周期性控制消息的应用，激活独立的信令字节应用，真实中心节点广播低速 VS 信令消息（几分钟一条 VS 信令消息），因此开销低，通过在真实中心节点实现智能化管理可以使这种开销达到最低。寻找 VS 的信令字节嵌入法也可以用来控制 VS 对其相邻节点的可见度。

信令字节包含一个 VS-TTL 组成域，用于说明该 VS 被广播的范围（转发跳数）。VS-TTL 设为 l ，则表示允许该 VS 最多 l 跳范围内的节点使用 SIPHON 的过载流量管理服务。 l 越大，允许使用本地 VS 的节点越多，但是增大 l 不一定会得到更好的网络性能。首先，只采用大值 l 时，各个节点的分组需要传递通过较长路径才能到达 VS，这种使用方法不会带来好处。

大 VS 广播范围导致 VS 周围附近拥塞机会的提高（每个 VS 可能产生类似于漏斗问题的微漏斗效应）。另外， l 越小，改道路径就越短，交付时延和能耗得到改善的节点极少。因此，需要确定各种条件下的 l 最佳值。

VS 和非 VS 对信令字节消息的处理方式各不相同。每种情形下的处理流程概述如下。真实中心节点若是没有配备第二部电台，则将广播的 SIPHON 控制分组（任何非数据分组均包含一个 SIPHON 信令字节）的 VS-TTL 设为 NULL；否则，即真实中心节点配有第二部电台，则将 VS-TTL 设为 l 。对于 VS 节点，任何输入控制分组若是嵌入了信令字节，则将该分组的转发节点作为下一个 SIPHON 转发跳；假如控制分组是通过第二部电台传输到达的，则将 VS-TTL 设为 l ，然后（以及在输入控制分组没有嵌入信令字节情形下）通过两个无线接口转发该控制分组。VS 若是通过其低功率电台接收到嵌入有 SIPHON 信令字节的控制分组，则将 VS-TTL 设为 NULL，然后不对其相邻区域广播自己存在的消息；这种 VS 没有通过其次网络到达真实中心节点的路径，因此其他节点通过这种 VS 节点来转发分组不会得到额外好处。但是，SIPHON 协议定义允许与配有两部电台的任意真实中心节点无连接的 VS 图通过其次网络来传输流量。对于非 VS 节点，任何输入控制分组若是嵌入了信令字节，则将 VS-TTL 设为大于 0 的数，然后将该分组的转发节点作为一个 VS 相邻节点，并递减 VS-TTL，然后（以及在输入控制分组没有嵌入信令字节情形下）转发该控制分组。

存在一个 VS 相邻节点表示一个 VS 位于该相邻区域内，以及通过这个特定相邻节点可达这个 VS。传感器节点运用这个规程维护一张 VS 可达相邻节点表。由于 VS 只占网络的一小部分，因此 VS 可达相邻节点表中常常只有一个相邻节点，所以维护该表的存储开销可忽略不计。在很多情形下，可以使用一比特表示路由表中的每个相邻节点条目。

8.2.3 SIPHON拥塞检测

精确而高效的拥塞检测指出传感器使用其已发现 VS 的合适时间，因而在 SIPHON 框架体系结构中起着重要作用。下面描述两种拥塞检测控制与 VS 基础设施激活技术：①节点初始化拥塞检测；②真实中心节点初始化事后拥塞检测。下面将介绍这两种技术及其在 SIPHON 中的运用。

1. 节点初始化拥塞检测

在 SIPHON 中，利用 CODA 机制确定节点当前的本地拥塞程度。但是，在本地信道载荷接近或者超过信道吞吐量的理论上限值或者缓存器缓存容量高于某个值后测量拥塞程度时，VS 可见度范围内的传感器节点激活改道算法，利用 VS 来改变相邻区域外的特定流量（比如数据脉冲、优先级化的流量等）的传输路径。为了更好地抑制漏斗效应，有必要尽早改变漏斗传输中的过载事件流的传输路径。但是，为了不影响网络中相关数据的累积（累积是漏斗中的最有效深度），改变漏斗中随后流量的传输路径是有益处的。为了平衡，最好在漏斗中很可能发生拥塞之前的某个位置改变数据的传输路径。

2. 事后拥塞检测

作为节点初始化拥塞检测方法的一个备用方法，考虑通过在真实中心节点发生的拥塞干

扰进行事后激活 VS 基础设施。真实中心节点是漏斗数据收集点，可以对事件数据质量以及所测应用逼真度进行智能化监视，以及只在所测应用逼真度低于某个门限值的时候才初始化 VS 信令。采用这种方法后，只是在主低功率无线网络中检测到拥塞情况或者应用逼真度下降问题之后才使用 SIPHON 服务。因此，这种方法处理网络短暂拥塞深度问题的能力有限，但是当拥塞发生地点离真实中心节点较近时，该方法可能很精确。这种技术的优点是不需要每个节点的低层拥塞检测支持。为了真实中心节点及时发送信令，利用无拥塞次无线网络（要求是连通的）来广播控制消息。由于事后拥塞检测法中的流量传输以真实中心节点的性能测试结果为基础而不是以网络拥塞程度为基础，所以这种技术的另外一个优点是避免了过早而仓促的流量传输，特别是在采用全网累积（如 AIDA）的时候。

8.2.4 改变流量的传输路径

在网络层头中使用一个改道比特，激活 SIPHON 流量改道。采用以下两种方法来设置改道比特：①按需改道，只是在检测到拥塞问题之时才设置改道比特；②始终改道，始终设置改道比特。基本改道机制如下：一个传感器节点接收到一个改道比特被置位的分组后，将其转发给自己的 VS 相邻节点，该分组通过这个 VS 相邻节点最终将到达一个 VS。假如所收分组的改道比特未被置位，那么该分组的传输路由遵从低层数据分发/路由协议确定的路径。

一个 VS 接收到一个被改道的分组后，将其转发给如下相邻节点：最近通过这个相邻节点接收到一条嵌入了信令字节的控制消息。这种控制消息可以通过 VS 的主无线接口或者次无线接口传递到达。在最佳情形下，所有 VS 通过次无线覆盖网络与真实中心节点连接，在一条快速路径上将真实中心节点周围的，通过该 VS 的所有分组转发给真实中心节点。次网络分割后，次网络分割部分中与真实中心节点最近的 VS 必须将其周围的所有分组回传给主网络，特别是传递给在寻找过程中已经确定的那个传感器节点，然后按照默认路由从该节点将分组转发给真实中心节点。

实验研究指出：采用低功率电台的 WSN 经常遇到链路质量随着时间和位置而强烈变化的问题。为了确保通过 VS 基础设施进行流量传输不会引起网络主要分组转发服务的质量下降，只使用链路质量好的相邻节点给 VS 转发分组。很多路由协议（比如 MultiHopRouter）维护一张相邻节点表，该表包含特定选择的相邻节点集的、需要连续更新链路质量估计。VS 可见度范围内一个传感器节点在转发事件分组时检测到拥塞后，根据本地策略作出特定数据分组类型改道传输决策。

作为 SIPHON 中流量改道的通用策略，一个备用下一个转发跳相邻节点（改道）的链路质量估计必须在当前选定下一个转发跳的链路质量估计的 15% 以内（下限），否则不适合使用这个 VS。若是满足改道策略参数，那么拥塞节点将数据分组路由头中的改道比特置位，然后将该数据分组转发给一个从其本地相邻节点表中选出的 VS 相邻节点。遵从合适策略，允许采用 VS 基础设施提高应用数据逼真度，绕过漏斗拥塞，采用本地 VS 的低质量链路不会导致分组丢失达到难以接受的程度。

VS 存在一些缺点，而且可能需要较高带宽交付数据。只是假如所用路由性能参数（比如跟数据交付网络路径有关的时延、丢失率）对强化服务特性敏感，那么通过 VS 进行流量传输才可能对主网络和次网络的路由协议操作稍有影响。例如，数据中心分发协议（比如定向扩散）能够选择经验质量好、动态适应网络状态变化的路径。因此，数据中心路由协议是

时延敏感协议，其路由决策稍受流量传输影响。

8.2.5 次网络中的拥塞

流量传输服务是 CODA 等拥塞控制技术的补充技术，因此可以在主网络和次网络上与这些拥塞控制技术同时运行。当次网络也发生过载时，通过 VS 的流量改道就没有意义了。因此，VS 总是监视其主信道和次信道上的拥塞程度，并且当主网络或者次网络发生过载时不对外广播存在拥塞问题。对于 IEEE 802.11 电台（在 SIPHON 实验中采用），Murty 等人^[21]提出了一个碰撞比特误码率计算算法，用于预测拥塞和动态调整 MAC 参数，优化吞吐量。这里采用这个技术对 SIPHON IEEE 802.11 次网络拥塞进行可靠检测，迫使 VS 根据检测出的拥塞程度抑制过载流量管理服务或者缩小其服务范围。

当主网络和次网络都发生过载时，两个网络的拥塞程度最终会高于某个门限值，此时触发 CODA 反压机制（即系统退回到传统的源节点和转发节点速率控制，减轻拥塞）。一般地，VS 能够通过两部不同电台，在具有不同特点（例如衰落、吞吐量、时延等）信道上同时发送和接收分组。

8.2.6 虚拟中心节点开销分析

仿真结果和实验结果说明使用 SIPHON 能够改善信道载荷不断提高的网络性能。下面介绍使用 SIPHON 后引入的开销。VS 按照一直开机的方式工作，比按需方式工作更快地耗尽能量，这与主网络和次网络的无线接口的特性密切相关。运用简单模型（能够捕捉测试床上分组交付的能耗），分析 VS “一直开机”与“按需开机”问题以及在测试床上使用 VS 的开销。在分析 SIPHON 开销时采用一个叫做“VS 使用开销比率”的指标，用于标准化采用 VS 将分组交付给真实中心节点的能量开销与只通过主网络交付同样数量分组给真实中心节点的能量开销之比。能量模型捕捉测试床上分组交付的发送能耗和接收能耗，包括分组发送的协议特性（比如采用 TinyOS/Mica2 和 IEEE 802.11B MAC 应答消息时的平均 CSMA 退避时间）。“VS 使用开销（VS Usage Cost, VSUC）”定义如下：

$$VSUC = E_{\text{delivery,VS}} / E_{\text{delivery,noVS}} \quad (8-17)$$

式中

$$E_{\text{delivery,noVS}} = \lambda N h_{\text{AVG,PS}} (e_{\text{mote,tx}} + e_{\text{mote,rx}}) + (1 - \lambda) N e_{\text{mote,tx}}$$

$$E_{\text{delivery,VS}} = \alpha E_{\text{delivery,noVS}} + (1 - \alpha) N \left\{ (1 - \lambda) e_{\text{mote,tx}} + \lambda \left[e_{\text{wifi,rx}} + e_{\text{wifi,tx}} + h_{\text{avg,VS}} (e_{\text{mote,tx}} + e_{\text{mote,rx}}) \right] \right\}$$

式中， N 表示各个源节点产生的分组数量之和， λ 表示 N 中被交付给真实中心节点的分组数量所占比例， $h_{\text{avg,PS}}$ 表示从源节点到达真实中心节点的平均转发跳数， $e_{\text{mote,tx}}$ 、 $e_{\text{mote,rx}}$ 分别表示 Mica2 MAC 层和物理层的发送能量和接收能量， α 表示 Mote WSN 单独交付的分组所占比例， $h_{\text{avg,VS}}$ 表示通过 IEEE 802.11B 网络交付的源节点分组到达 VS 之前所通过的 Mote WSN 的平均转发跳数， $e_{\text{wifi,tx}}$ 、 $e_{\text{wifi,rx}}$ 分别表示 IEEE 802.11B MAC 层和物理层的发送能量和接收能量。

N 由特定源节点分组发送速率决定， λ 、 α 、 $h_{\text{avg,PS}}$ 、 $h_{\text{avg,VS}}$ 从实验数据记录中得到。CC1000 防火墙设置的发射功率为 41.31 mW，接收功率为 21.09 mW。分组时间包括应用分组长度、前

导、启动符号，速率 19.2 kb/s 的分组时间为 27.5 ms。假定在 Mote WSN 中只需要初始 CSMA 退避时间，假定每次发送的平均退避时间为 3.125 ms。运用这些数据计算出 $e_{\text{mote,tx}}=1\,201.889\,\mu\text{J}$ 、 $e_{\text{mote,rx}}=597.975\,\mu\text{J}$ 。IEEE 802.11 网卡的发射功率、接收功率分别为 1.4 W、1.0 W（见表 7-4）。包括平均退避时间、SIFS、DIFS、DATA、ACK 分组时间（包括 TCP/IP、MAC、PLCP 头），计算平均发送时间和接收时间，得到分组发送能量 $e_{\text{mote,tx}}=1\,184.55\,\mu\text{J}$ 、分组接收能量 $e_{\text{mote,rx}}=581.82\,\mu\text{J}$ 。这里的 IEEE 802.11B 网络没有使用 RTS/CTS。没有考虑在每个转发跳可能多个 Mote 传感器接收到同一个分组。但是，假定主网络预期节点密度高于次网络节点密度以及 Mote WSN 的计算接收开销高于 IEEE 802.11B 网络的计算接收开销，忽略这些接收开销则不支持 SIPHON。

图 8-7 给出了 4 个源节点组的 SIPHON 开销。从图 8-7 中看到：使用 SIPHON 的开销实际上低于只使用主网络的开销。因为 VS 可见度范围设为 1，VS 布置在源节点附近，所以采用 1 个 VS 和 2 个 VS 的平均转发跳数距离小于没有 VS 的平均转发跳数距离。由于使用 TinyOS/Mica2 和 IEEE 802.11B 的每个分组能量开销类似，所以从源节点到达真实中心节点的平均转发跳数以及小 α （由于链路衰减）主导能量计算，因此采用 SIPHON 后能量开销得到改善。但是，在 4 个源节点网络中，即使大部分分组通过次网络传递，IEEE 802.11B 信道载荷仍然很高，这就意味着大部分时间空闲。在模型中综合空闲时间的能量开销，导致“一直开机”工作方式的能量开销大幅度增大。

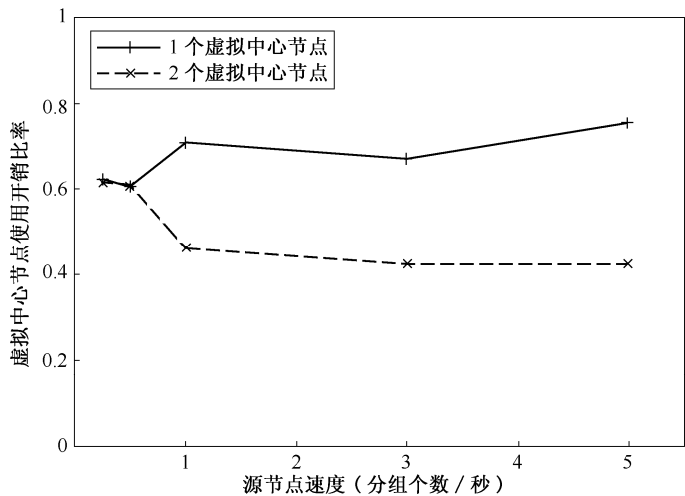


图 8-7 SIPHON 性能

参 考 文 献

[1] S. D. Servetto and G. Barrenechea. Constrained random walks on random graphs: routing algorithms for large scale wireless sensor networks. in Proc. ACM WSNA, Atlanta, GA, Sep.2002, pp.12–21.

[2] E. Shih et al.. Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks. in Proc. ACM MOBICOM, Rome, Italy, Jul. 2001, pp.272–286.

- [3] K. Sohrabi, B. Manriquez, and G. Pottie. Near-ground wideband channel measurements. in Proc. IEEE Vehicular Technology Conf.(VTC), vol. 1, New York, 1999, pp.571–574.
- [4] J. Zhao and R. Govindan. Understanding packet delivery performance in dense wireless sensor networks. in Proc. ACM SENSYS, 2003, pp.1–13.
- [5] M. C. Vuran, O. B. Akan, and I. F. Akyildiz. Spatio-temporal correlation: theory and applications for wireless sensor networks. Comput. Netw. J., vol.45, no.3, pp.245–261, Jun. 2004.
- [6] Y. Sankarasubramaniam, O. B. Akan, and I. F. Akyildiz. ESRT: Event-to-sink reliable transport in wireless sensor networks. presented at the ACM MobiHoc, Annapolis, MD, Jun.2003, pp.177-188.
- [7] O. B. Akan, and I. F. Akyildiz. Event-to-Sink Reliable Transport in Wireless Sensor Networks. in Proc. IEEE/ACM TRANSACTIONS ON NETWORKING, VOL.13, NO.5, pp.1003-1016, OCTOBER 2005.
- [8] D. Bertsekas and R. Gallager. DATA NETWORKS, second edition. Prentice Hall, Upper Saddle River, New Jersey, 1992.
- [9] J. Li, C. Blake, D. D. Couto, H. Lee, and R. Morris. Capacity of ad hoc wireless networks. In Proc. of the Seventh Annual International Conference on Mobile Computing and Networking, pages 61–69, July 2001.
- [10] C-Y. Wan, S. B. Eisenman, and A. T. Campbell. CODA: COngestion Detection and Avoidance in Sensor Networks. In Proc. of the 1st ACM Conf. on Embedded Networked Sensor Systems, pages 266-279. Los Angeles, Nov 5-7 2003.
- [11] C-Y. Wan, A. T. Campbell, and J. Crowcroft. A Case for All-Wireless Dual-Radio Virtual Sinks. In Proc. of the 2nd ACM Conf. on Embedded Networked Sensor Systems, pages 267-268. Baltimore, Nov 3-5 2004.
- [12] A. Arora, et al.. ExScal: Elements of an Extreme Scale Wireless Sensor Network. In Proc of the 11th IEEE Int'l Conf. on Embedded and Real-Time Computing Systems and Applications. Hong Kong, Aug 17-19 2005.
- [13] J. Zhao and R. Govindan. Understanding Packet Delivery Performance in Dense Wireless Sensor Network. In Proc. Of the 1st ACM Conf. on Embedded Networked Sensor Systems, pages 1-13. Los Angeles, Nov 5-7 2003.
- [14] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman. Infrastructure Tradeoffs for Sensor Networks. In Proc.of the 1st ACM Int'l Workshop on Wireless Sensor Networks and Applications, pages 49-58. Atlanta, Sep 2002.
- [15] E. Shih, P. Bahl, and M. J. Sinclair. Wake on wireless: An Event Driven Energy Saving Strategy for Battery Operated Devices. In Proc. of the 8th Annual Int'l Conf. on Mobile Computing and Networking. Atlanta, GA, Sep 2002.
- [16] M. Yarvis, N. Kushalnagar, H. Singh, A. Rangarajan, Y. Liu, and S. Singh. Exploiting Heterogeneity in Sensor Networks. In Proc. of IEEE INFOCOM. Miami, FL, Mar 2005.
- [17] F. Xue and P. R. Kumar. The Number of Neighbors Needed for Connectivity of Wireless Networks. Wireless Networks, 10(2):169-181, Mar 2004.

- [18] Stargate datasheet. <http://www.xbow.com>.
- [19] A. Woo and D. Culler. Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks. In Proc.of the 1st ACM Conf. on Embedded Networked Sensor Systems, pages 14-27. Los Angeles, Nov 5-7 2003.
- [20] Tinyos homepage. <http://webs.cs.berkeley.edu/tos/>.
- [21] R. Murty, E. H. Qi, and M. Hazra. An Adaptive Approach to Wireless Network Performance Optimization. In Wireless World Research Forum (WWRF11 Meeting). Oslo, Norway, Jun 10-11 2004.
- [22] A. Basu, B. Boshes, S. Mukherjee and S. Ramanathan. Network Deformation: Traffic-Aware Algorithms for Dynamically Reducing End-to-end Delay in Multi-hop Wireless Networks. In Proc. of the 10th Annual Int'l Conf. on Mobile Computing and Networking, Philadelphia, USA, Sep 2004.
- [23] Chieh-Yih Wan. A Resilient Transport System for Wireless Sensor Networks. Ph.D Thesis, Columbia University, USA, 2005.
- [24] Y. G. Iyer, S. Gandham, and S. Venkatesan. STCP: A Generic Transport Layer Protocol for Wireless Sensor Networks. Proc. IEEE ICCCN 2005, San Diego, CA, Oct. 17–19, 2005.
- [25] J. Postel. RFC 768: User Datagram Protocol. August 1980.
- [26] K. Ramakrishnan and R. Jain. A Binary Feedback Scheme for Congestion Avoidance in Computer Networks. ACM Transactions on Computer Systems, May 1990.
- [27] Philip Levis, Nelson Lee, Matt Welsh and David Culler. TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications. Proceedings of 1st International Conference on Embedded Networked Sensor Systems, 2003.
- [28] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, V. Paxson. RFC 2960: Stream Control Transmission Protocol. Network Working Group, Oct. 2000.
- [29] S. Floyd and V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. IEEE/ACM Transactions on Networking, August 1993.
- [30] S. Tilak, N. B. Abu-Ghazaleh and W. Heinzelman. Infrastructure Tradeoffs for Sensor Networks. In Proc. of WSNA02, Atlanta, Georgia, USA, Sept. 2002.
- [31] V. Jacobson. Congestion Avoidance and Control. Proceedings of the ACM SIGCOMM Symposium, August 1988.

第 9 章 无线传感器网络逐跳可靠传输协议

拥塞检测与控制技术是传输层重要任务之一。第 8 章介绍了端到端拥塞检测与控制技术 (CODA)，本章首先介绍一种逐跳拥塞检测与控制技术 (FUSION)，然后介绍两个逐跳可靠传输协议 (PSFQ 和 GARUDA)。

9.1 合成拥塞控制技术 (FUSION)

提供极少发生拥塞问题的 WSN 是非常困难的。WSN 交付各种类型的流量，从简单周期性报告到不可预测的、由所感知到的外部事件触发的突发消息。因为无线信道随着时间而变化且常常剧烈变化，在不同无线“链”上同时进行数据传输互相作用、互相影响，从而导致信道质量不仅与噪声有关而且与流量密度有关，因此，即使在周期性流量模式和简单网络拓扑下，WSN 仍然会发生拥塞问题。此外，传感器的入网与退网，报告速率的变化，都会引起网络原先不拥塞部分变得拥塞。当感知事件引起突发消息时，更加可能发生拥塞。

在传统有线网络和蜂窝无线网络中，缓存器溢出和时延增大是拥塞的征兆。通常综合利用端到端速率自适应技术和网络层丢失与信令技术来确保有线网络和蜂窝无线网络不会因拥塞而崩溃。除了缓存器溢出之外，WSN 拥塞的一个主要征兆是网络其他区域发送的流量增大导致无线信道质量恶化。因为无线“链路”不像有线信道或者蜂窝无线链路那样物理上相互屏蔽、相互隔离，所以传输通过网络一个给定区域的流量对网络其他部分的信道质量和丢失率会产生不利影响。无线信道质量差且具有时变特性、不对称通信、隐含终端导致调整好的流量难以交付；在流量载荷作用下，多跳 WSN 严重影响分组的多跳传输，导致极大的不公平性。

9.1.1 拥塞崩溃的症状

图 9-1 所示的结果来自美国麻省理工学院计算机科学与人工智能实验室报告的室内 Mica2 WSN 测试床的实验结果。每个节点以恒定速率产生数据，其他传感器通过多跳传递将数据转发给中心节点。随着载荷的增加，丢失率迅速提高。图 9-1 (a) 表示各种载荷下的全网丢失率。将无线信道误码引起的丢失率和缓存器溢出引起的丢失率分离开，那么从图 9-1 (a) 中可以看到无线信道误码对丢失率起主导作用，且随着载荷的增加而迅速提高。丢失率急剧增大是拥塞崩溃的两个主要症状之一。

拥塞崩溃的第二个主要症状是由于中心节点一跳远节点产生的流量而使得网络大部分饥饿，图 9-1 (b) 表示这种现象。假定一个节点产生的分组被中心节点所接收到的百分比为 p ，那么互补累积分布函数 (Cumulative Distribution Function, CDF) 表示至少将其 $p\%$ 的分组交付给中心节点的传感器所占比例。从图 9-1 (b) 中看到：随着载荷的增加，互补 CDF 下降 (即节点减少)，不成比例地占用大部分网络带宽。

拥塞崩溃对 WSN 能量效率带来极为不利的影响，如图 9-1 (c) 所示。从图 9-1 (c) 中看到，当载荷高于拥塞点后，同样能量下能够发送的比特数很少。从边沿节点发送给中心节点的数据只会被丢掉，无法传递到达中心节点，因此造成能量浪费，将这种现象称为活锁 (Livelock)。

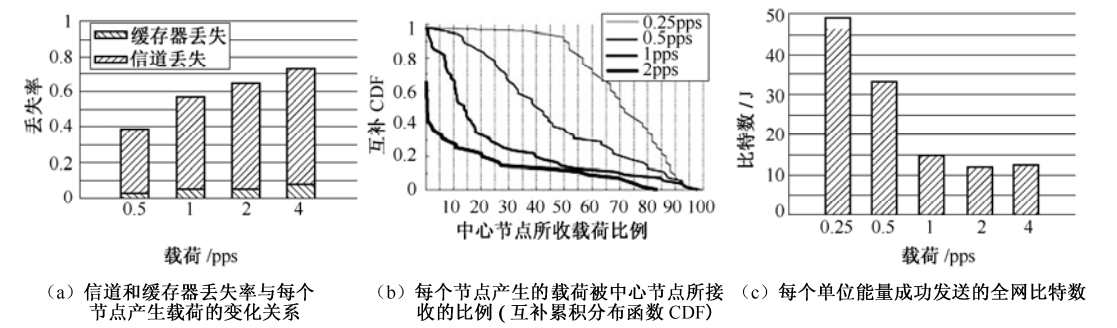


图 9-1 WSN 测试床在缺乏拥塞控制策略时的拥塞崩溃

美国麻省理工学院计算机科学与人工智能实验室研究和开发了一种叫做合成拥塞控制技术 (FUSION)。FUSION 由三种拥塞控制技术综合而成：第一种技术是逐跳流量控制技术，节点通过反压互相发送本地拥塞信号，防止节点在其下行节点队列发生溢出而导致其上行节点发送只会被丢掉时做出无用发送，降低分组丢失率；第二种技术是源速率限制技术，用于减轻对其数据传递必须通过许多转发跳才能到达中心节点的源节点的严重不公平问题；第三种技术是 MAC 层优先级化技术，使拥塞节点的共享信道访问权优先于非拥塞节点，避免发生缓存器溢出问题。每种技术都能在一定程度上减轻拥塞问题，但是将三种技术综合在一起使用，合成技术能够大幅度地改善网络效率、公平性、信道丢失率，从而大幅度地减轻网络拥塞造成的不利影响。

在无线环境中应用这三种技术是很困难的。第一，发送节点和接收节点上同时存在信道竞争问题，即使在室内，信道竞争也是一个严重问题，因为无线传播反射引起传播不稳定，导致两个似乎不相交节点集之间相互干扰；第二，信道利用率和公平性之间自然需要折中平衡，允许大流量节点发送，则必然以最大程度竞争给这些节点分配带宽；第三，无线信道本来就是有损信道，因此对数据流进行分布式控制更加富有挑战性。下面进一步分别描述这三种拥塞控制技术。

9.1.2 逐跳流量控制

逐跳流量控制已经应用于有线局域网、有线广域网以及 WSN。在合成拥塞控制技术中，每个传感器在其发送的每个分组的分组头中设置一个拥塞比特。利用无线媒介的广播特性，通过每次发送将拥塞信息反馈给相邻区域中的所有传感器节点。这就意味着拥塞信息反馈不需要使用直接控制消息，直接拥塞控制消息需要占用一部分带宽。逐跳流量控制由两部分组成：拥塞检测和拥塞减轻。首先讨论两种拥塞检测方法：队列占用法和信道采样法。

一种检测拥塞的简单方法就是监视传感器的队列长度。假如输出队列可用空间部分进入高水位线 α 以下，则将输出分组的拥塞比特置位；否则将输出分组的拥塞比特清零。

另外一种拥塞检测法就是 CODA 的信道采样检测法。当一个分组等待发送时，传感器按

照固定间隔时间采样信道状态。根据信道忙的时间长度计算利用率因子。假如利用率高于某个门限值，则将输出分组的拥塞比特置位；否则，将输出分组的拥塞比特清零。进一步的详细描述请参阅第 8 章。

拥塞减轻是一种机制。给定相邻区域内的节点利用拥塞减轻机制抑制其发送，防止下一个转发跳节点发生溢出。一个传感器若是旁听到其父节点发送来的一个分组的拥塞比特被置位，则停止转发数据，以便允许父节点将其队列中分组全部发送完毕。若是不采用这种反馈机制，则在流量波传递通过网络时分组缓存器可能很容易发生溢出。假如一条路径持续拥塞，那么逐跳反压最终到达源节点，允许应用层流量控制抑制源节点速率。

当需要多跳传递反压时存在一个有关拥塞状态通信问题。当父节点设置拥塞比特时，子节点停止发送，因此，子节点在发生拥塞时就无法通知自己的子节点。解决方法是：允许拥塞节点一旦旁听到其父节点发出的一个分组设置了拥塞比特就立即发送一个分组；也可以拥塞节点每当接收到一个分组就发送一个分组，补偿子节点旁听不到表示拥塞的分组。

9.1.3 速率限制

由于信道质量和承载载荷的不断变化，网络中任何点都可能发生拥塞。拥塞点常常引起噪声平面的升高、分组交付率的急剧下降。随着网络直径的增大，假如传递流量由于拥塞而被丢掉，那么这就变成一个越来越严重的问题，这是因为 WSN 消耗了许多能量和带宽来发送多跳转发分组（活锁问题）。而且 WSN 的一种自然趋势是：交付离中心节点近的传感器产生的流量，牺牲离中心节点远的传感器产生的流量。源节点速率限制就是解决噪声平面升高问题和不公平性变强问题。

速率限制工作原理如下：假定所有传感器产生相同流量载荷、路由树基本对称，处理速率变化的一种较好通用方法是要求源节点发送其流量产生速率。为了简单起见，采用全被动法，即通过传递流量监视来确定源节点速率。每个传感器旁听其父节点转发的流量，据此估计其路由通过该父节点的源节点总个数 N ；然后使用标记斗（Token Bucket）法调整每个传感器的发送速率。传感器每当旁听到其父节点转发了 N 个分组就累加一个标记，直到累加到最大标记数量为止。只有当传感器的标记累加计数器大于零的时候，才允许该传感器发送，并且每发送一次就将其标记数减 1。这种方法限制传感器以其每个子节点的相同速率进行发送。

9.1.4 MAC层优先级化

传感器运用上述网络层机制能够对拥塞问题做出反应和处理，但是对拥塞问题的反应速度不能保证总是足够快的，从而不需要 MAC 层的支持就能够防止缓存器丢失。CSMA 对拥塞控制具有辅助作用。

标准 CSMA 对每个传感器具有相同的发送竞争成功概率。但是，在拥塞期间，若采用标准 CSMA，则由于拥塞节点不能迅速将拥塞控制反馈给相邻节点，因而可能引起性能下降。例如，考虑几个传感器通过一个公共父节点来转发分组。通常，父节点只有在其一半相邻节点发送之后才能够访问信道。但是，因为父节点拥塞，所以父节点可能没有足够存储空间来存储其子节点转发来的分组，没有选择余地，只得丢掉子节点转发来的分组。因此，拥塞节点必须有访问无线媒介的优先权。

为此，使每个传感器随机退避时间长度（在每个发送周期之前）作为其本地拥塞状态的函数。拥塞传感器的退避窗口等于非拥塞传感器退避窗口的 1/4，使拥塞传感器赢取竞争的机会更高，允许将输出队列中分组全部发送完毕，提高拥塞控制信息传播到传感器整个相邻区域中的可能性。

两个不在其无线覆盖范围内的发送节点同时对一个公共接收节点发送时就会发生隐含终端问题。减少隐含终端间碰撞的一种方法就是在通信前互相交换 RTS/CTS 控制分组。尽管 RTS/CTS 也可能碰撞，并且有些不会碰撞的发送可能会停止进行，但是 RTS/CTS 交互排除了大部分数据分组的碰撞。当数据分组长度远大于控制分组时，RTS/CTS 交互带来的开销是值得的。但是在 WSN 中，数据分组通常较小，并且根据研究文献报告在有些平台上进行 RTS/CTS 交换会增加 40% 的开销。

采用如下策略减轻树状拓扑中的隐含终端问题：一个节点旁听到其父节点发送完一个分组后，等待一个分组时间及一个保护间隔时间，以避免可能的隐含终端与其祖父节点碰撞。

9.1.5 应用自适应

应用在预防拥塞中起着重要作用。当网络栈没有准备接收额外数据时，通过发送失败来通知应用，然后一直等到应用做出适当响应为止。有些应用只是等待网络栈重新准备好接收数据；而有些应用则通过 AIMD 控制器或者类似机制来调整其发送速率。通常，应用只允许在网络栈中立即保存少量分组，防止本地流量成为缺乏传输路由的流量。

9.2 慢分发、快提取可靠传输协议（PSFQ）

对传感器组进行空中重复编程或者重新分配任务是一种 WSN 应用，这种 WSN 应用要求底层传输协议支持可靠数据交付。现在，WSN 是针对特定应用的，通常采用有线方式低成本、高效率地执行某个特定任务。随着 WSN 应用的不断增多，将需要建立功能更加强大、更加通用的硬件和软件环境，能够对传感器进行重复编程或者重新分配任务，使传感器能够完成多种任务。这种通用传感器能够服务新应用类型和将要出现的应用类型。这种系统正在出现，比如，伯克利 Mote 传感器能够从网络上接收代码段，并且在传感器重新分配任务前，将各个代码段组合成一个全新的执行代码存入 E²PROM 二级存储器中。

开发 WSN 可靠传输协议面临许多挑战。例如，在重新分配任务应用中，可能需要对特定传感器组进行重新编程（如在灾后恢复区），因此需要寻址传感器组，对其加载新的二进制代码，然后切换到新的应用中，要求所有这些操作都是可控的。随着网络中传感器节点不断增多，对传感器节点重新分配任务变得愈加富有挑战性。可能需要采取可控、可靠、强壮、及时、可扩展的方式对数百个甚至数千个传感器节点重复编程，那么传输层如何才能正确支持重新分配任务应用呢？这种可靠传输协议必须是微小而能量高效的，以便于在低端传感器节点（如伯克利 Mote 传感器系列）上实现；这种可靠传输协议能够以高效而强壮方式将应用与 WSN 不可靠特性隔离开。WSN 经历的误码率变化范围大，因此可靠传输协议必须能够在这种条件下将数据可靠交付给大量传感器组。

为此，美国哥伦比亚大学开发了一个 WSN 可靠传输协议，即慢分发、快提取（Pump Slowly, Fetch Quickly, PSFQ）协议。由于 WSN 的应用特定性，所以很难将一个特定可靠

传输协议优化成每个应用都适用的协议。PSFQ 代表一种简单方法，具有最低路由基础设施要求（不同于 IP 多目标路由要求）、最低程度信令，因此降低了数据可靠性通信开销，对高误码率反应迅速，在高误码率条件下也仍然能够成功操作。

9.2.1 PSFQ协议概述

支持 PSFQ 设计的关键思想是：以相对较慢的数据速率分发源节点的数据（称为慢分发），而正在经历数据丢失的节点主动从其本地直接相邻节点中提取（恢复）已丢失的数据片（本地恢复，称为快提取）。节点接收到的序列号大于预期序列号，则检测到已丢失消息，因此触发提取操作（即作为 PSFQ 基础的能量高效否定应答系统）。PSFQ 背后的动机是通过直接相邻节点间的本地数据恢复，实现宽松时延范围、同时丢失恢复开销最低。

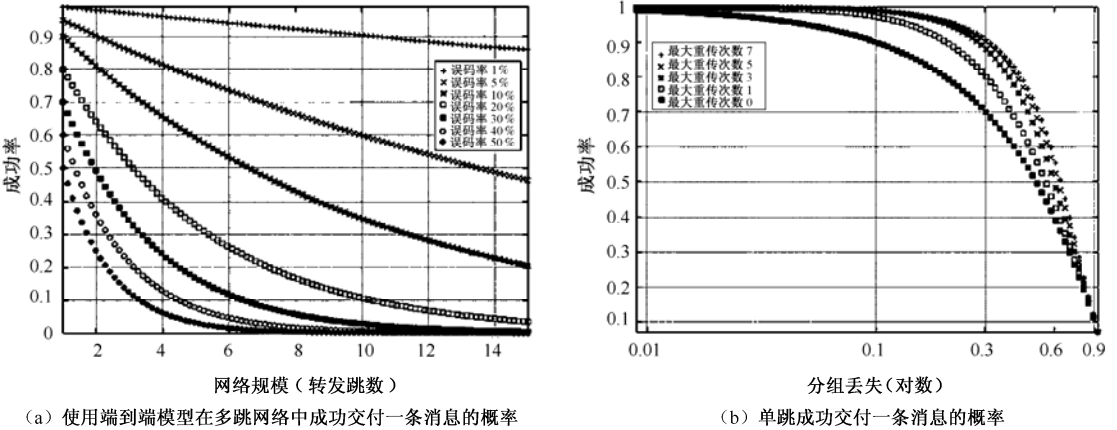


图 9-2 成功交付一条消息的概率

1. 逐跳误码恢复

对于传统端到端误码恢复机制，只是最终目的节点负责丢失检测和请求重传。端到端恢复机制的最大问题是不得不处理传输媒介的物理特性。WSN 通常在苛刻无线环境下工作，依靠多跳转发技术交换消息。误码随着转发跳数的增加而呈指数累加，因此很可能发生分组丢失和重新订购分组。为了简单说明这个问题，假定无线信道的分组丢失率为 p ，那么经过 n 个转发跳成功交换一条消息的概率为 $(1-p)^n$ 。图 9-2 (a) 用数据说明这个问题。图 9-2 (a) 给出了成功率与网络规模（转发跳数）的变化曲线：对于较大规模网络，当误码率高于 10% 时，在有损链路环境中使用端到端恢复机制几乎不能交付单条消息。参考文献[11]和[12]指出：在密集 WSN 中经常遇到 10% 或者更高的误码率。在很多情形下，比如军事应用、工业过程监视、灾后重建，误码率更高。这些事实说明端到端误码恢复机制不是 WSN 可靠传输的良好候选机制。

因此，PSFQ 采用逐跳误码恢复机制，中间节点也负责丢失检测和恢复丢失的数据片，因此在逐跳基础上进行可靠数据交换。逐跳误码恢复实质上是 将多跳转发操作分割成一系列的单跳发送过程，排除误码累加。因此，与端到端法相比，逐跳法可扩展性更好，抗误码能

力更强，同时降低了分组重新订购的概率。

2. 提取/分发的关系

对于否定应答系统，数据交付时延依赖成功交付预定的重传次数。为了降低时延，必须使“一个可控时间帧”内成功交付一个分组的概率最大。一种直观方法是：在下一个分组 $i+1$ 传递到达之前尽可能多次重传分组 i （因此提高了成功交付的概率），也就是说，在新分组传递到达之前清除接收节点（比如中间传感器节点）的队列，以便维持短队列长度，从而降低时延。但是，确定最佳重传次数、平衡成功交付率（即在一个时间帧内成功交付单条消息的概率）同时不能在重传上浪费过多能量，不是一个能够轻易解决的问题。为了研究和证明这个设计决策是正确的，分析一个与该机制近似的简单模型。假定分组丢失率 p 在可控时间帧内恒定不变，那么在否定应答系统中，在最大重传次数为 k 的条件下，在两个节点间成功交付一个分组的概率的递推公式为

$$\begin{aligned} &(1-p) + p \times \Omega(k) \quad (k \geq 1) \\ &\Omega(k) = \Phi(1) + \Phi(2) + \cdots + \Phi(k) \\ &\Phi(k) = (1-p)^2 \times [1-p-\Phi(1)-\cdots-\Phi(k-1)], \quad (\Phi(0)=0) \end{aligned} \tag{9-1}$$

式中， $\Omega(k)$ 表示 k 次重传内成功恢复一个丢失数据片的概率， $\Phi(k)$ 表示第 k 次重传成功恢复一个丢失数据片的概率。

根据式 (9-1)，可绘出单跳成功交付一条消息的概率与分组丢失率的变化曲线[见图 9-2 (b)]，说明了增加重传次数对提高成功交付率的影响。从图 9-2 (b) 中看到：信道误码率位于 0%～60%之间时的成功交付率有明显提高，但是，进一步增加重传次数带来的交付率改善会迅速下降，当重传次数大于 5 时带来的交付率改善可忽略不计。这个简单分析意味着 PSFQ 分发操作定时器与 PSFQ 提取操作定时器之最佳比率约等于 5。

3. 多模操作

在否定应答系统中，假如连续转发较大序列号分组，则本地丢失事件会传播给下行节点。丢失事件传播会造成严重的能量浪费，因为一个丢失事件会触发误码恢复操作，试图迅速从直接相邻节点提取已丢失分组，但是其相邻节点（下行节点）却没有该丢失分组。因此，丢失分组不能恢复，有关提取操作的控制消息是浪费。所以必须确保中间节点只转发具有连续序列号的消息。

要求使用数据存储器来缓存消息，确保按照序列号连续顺序转发数据、从下行节点完整恢复丢失数据。对于重新分配任务应用，要求存储器能够保存全部代码段。PSFQ 分发机制只预防丢失事件传播以及预防从下行节点的不必要的提取操作，但是对于 PSFQ 的抗信道误码能力非常重要。通过丢失事件本地化、停止转发较大序列号消息（直到已经开始恢复操作为止），PSFQ 分发机制操作方式类似于存储转发方式。对于后者，中间节点只是在接收到完整文件之后才转发该文件。存储转发法实质上将多跳转发操作分割成一系列单跳发送过程，所以在高误码率环境下非常有效。

PSFQ 得益于存储转发法和分组交换之间的如下综合平衡：在低误码率条件下，当丢失分组能够迅速恢复时，PSFQ 分发操作按照多跳分组交换方式进行；当信道误码率高时，PSFQ 分发操作更像存储转发通信。因此，PSFQ 表现出多模通信特性，依据所遇到的信道条件在

分组交换和存储转发之间适度平衡。

4. PSFQ协议的组成

PSFQ 由三个协议功能组成：消息中继（分发操作）、中继初始化误码恢复（提取操作）、选择性状态报告（报告操作）。用户将消息发送到网络中，中间节点缓存消息，按照适当的时间安排中继消息，达到宽松的时延范围。中间节点维护一个数据存储器，利用所缓存的信息检测数据丢失，在必要时初始化误码恢复操作。用户获取有关分发状态统计数据，并将其作为随后决策（比如在进行空中重新任务分配/编程时修正切换到新任务的时间）的基础非常重要。因此，必须将反馈与报告机制综合到 PSFQ 中，使 PSFQ 是灵活的（即自适应环境）、可扩展的（即开销最低）。

下面针对重新分配任务应用描述 PSFQ 的主要操作（如分发、提取、报告）。在重新分配任务应用中，用户必须将控制字或者二进制代码段分发到目标传感器节点，重新对传感器节点子集进行任务分配。

9.2.2 PSFQ分发操作

PSFQ 不是路由协议，而是传输协议。倘若必须直接寻址一个特定节点，而不是寻址整个传感器组是规范操作，那么 PSFQ 可以工作在现有路由协议上面，支持可靠数据传输。为了支持任意点到任意点通信模式的可靠传输，PSFQ 位于路由层上面，在数据分组中使用单目标目的节点地址，而不是使用广播地址。为了支持逐跳误码恢复，需要旁听机制，将路由代理的分组复制给 PSFQ 代理，此时节点只需要按照路由算法的决策将分组转发给目的节点，参与 PSFQ 操作。用户节点（即中心节点）可以使用寿命时间（Time To Live, TTL）法和组地址滤波法控制任务重新分配操作范围。但是这些方法不提供精确的范围控制，这是因为在很多情况下，利用 TTL 限制不能精确定义预定接收节点。在中间节点创建和维护一个数据存储器，便于进行本地丢失恢复和按照序列号连续顺序进行数据交付。

PSFQ 分发操作对于控制代码段及时分发给所有目标节点、提供基本流量控制、使重新分配任务操作不会淹没 WSN 正常操作非常重要。这就要求正确安排数据转发时间。采用一个简单时间安排方案，该方案采用两个分发定时器 T_{\min} 和 T_{\max} ，描述如下。

用户节点按周期时间 T_{\min} 给其相邻节点广播一个分组，直到所有数据分片发送完毕为止。相邻节点接收到该分组后，检查其本地数据存储器，丢掉重复分组。假如是一个新分组，则将其存入数据存储器，将其 TTL 减 1；假如 TTL 不等于零并且序列号是连续的，那么 PSFQ 设置转发该分组的时间。延迟一段随机时间（位于 T_{\min} 和 T_{\max} 之间）后，将该分组转发给相邻节点（离源节点一跳或者多跳远）。在重新分配任务应用中，PSFQ 只是简单地重新广播该分组。按照这种发式，分组远离其源节点最多传递 TTL 个转发跳。当干扰节点间重新广播定时高度相关时，在广播操作下不适合使用 RTS/CTS 握手，因此转发前的随机时延对于避免传输碰撞是必需的。

T_{\min} 有几个考虑。第一，对于本地丢失恢复，需要提供时间缓存器。PSFQ 背后的主要动机之一是在可控时间帧内从直接相邻节点迅速恢复丢失分组。在这种意义下，在上行节点送来下一个数据片之前，按照 PSFQ 分发操作，节点必须至少等待 T_{\min} 后才会转发分组，因此有机会恢复丢失分组。第二，必须减少冗余广播。在密集布置的 WSN 中，常常是无线传输

覆盖范围内存在多个直接相邻节点。参考文献[9]指出：重播系统除了先前发送的已有覆盖范围，只能提供 0%~61% 的额外覆盖范围，而且假如一条消息已经被接收到 4 次，那么其额外覆盖范围低于 0.05%。利用 T_{\min} 后，节点在实际开始转发消息前有机会旁听其他重播节点广播的同一条消息。设置一个计数器，用于跟踪记录相同广播消息的接收次数。假如在所安排的转发时间之前接收到 4 次，则取消转发，因为相对于发送开销，转发该消息几乎没有什么益处（额外覆盖范围）。 T_{\max} 可以用来提供最后一个转发跳节点成功接收到整个文件最后一个数据片的宽松统计时延范围。假设采用如前所述主动提取操作在一个 T_{\max} 间隔内恢复任意丢失数据，那么时延范围 $[D(n)]$ 和 T_{\max} 之间的关系为（其中 n 表示一个文件的分片数量） $D(n) = T_{\max} \times n \times \text{转发跳数}$ 。

9.2.3 PSFQ提取操作

节点一旦检测到文件分片中出现序列号不连续问题，则立即进入 PSFQ 提取方式。提取操作是主动操作，请求相邻节点重传。只要检测到丢失，PSFQ 就使用“丢失累积”概念，即 PSFQ 对所有丢失消息采取批处理，尽量在一次提取操作中恢复所有丢失消息。

1. 丢失累积

由于衰落条件和其他信道损伤，数据丢失常常是时间相关的。经常发生成批的丢失（突发丢失）。PSFQ 累积丢失，提取操作处理各个丢失分组的“窗口”，而不是处理单个分组丢失。在密集 WSN 中，节点常常有多个相邻节点，有可能每个相邻节点在丢失窗口中只获得或者保留一部分丢失分片。PSFQ 允许从不同相邻节点恢复丢失窗口中的不同分片。为了减少相同分片的冗余重传，每个相邻节点等待一段随机时间后再发送分片。具有该数据且已安排重传时间的其他节点若是旁听到某个相邻节点的相同恢复分片，则取消其定时器。在恶劣无线环境中，可能发生连续丢失，包括重传分片和提取控制消息的丢失。因此，通常一个节点经过若干次丢失后，其接收到的消息中可能存在多处序列号不连续问题。在提取操作中累积多个丢失窗口，提高了成功恢复概率，在这种意义下，一个相邻节点只要接收到一条提取控制消息，就可以重传所有丢失分片。

2. 提取定时器

在提取方式下，一个节点主动对其直接相邻节点发送否定应答（Negative ACKnowledgement, NACK）消息，向其请求丢失分片。假如在周期时间 T_r ($T_r < T_{\max}$ ，定时器定义分发操作与提取操作的时间比率) 内没有接收到相邻节点的应答或者只恢复部分丢失分片，那么该节点按时间 T_r 周期重发 NACK（稍微随机化，避免相邻节点间同步），直到所有丢失分片恢复为止或者重传次数超过预定门限值而结束提取操作。安排发送的第一个 NACK 随机时延短，位于 $0 \sim \Delta$ 之间， $\Delta < T_r$ 。若是在发送 NACK 前旁听到另一个相邻节点发送 NACK 恢复相同丢失分片，则取消第一个 NACK（以便使重复的 NACK 和丢失分片很少）。因为 Δ 小，所以发生这种情况的概率相对较小。通常不能保证响应其他节点发送的 NACK 而进行的重传被取消其第一个 NACK 的那个节点所旁听到。参考文献[9]指出：在这种条件下，取消第一个 NACK 的节点最多有 40% 的机会接收到重传数据。但是，假如丢失分片没有恢复，那么节点最终会重

发 NACK。因此，这种方法是安全的，并且在特定折中下是有益的。

为了防止消息暴问题，绝不能传播 NACK，即相邻节点不能中继 NACK，除非同一个 NACK 的接收次数大于预定门限值，此时节点数据存储器不再保留通过 NACK 请求的丢失分片，此时 NACK 只被中继一次，NACK 范围再次被有效拓宽一个转发跳，提高丢失恢复概率。

接收到 NACK 的每个相邻节点检查丢失窗口域。假如在数据存储器中找到丢失分片，那么相邻节点安排一个应答事件(发送丢失分片)的发送时间，其随机延迟时间在 $(1/4)T_r \sim (1/2)T_r$ 之间。只要旁听到相同 NACK 请求相同的丢失分片，相邻节点就取消应答事件。当 NACK 中的丢失窗口包含多个分片需要重传时，或者当 NACK 中包含多个丢失窗口时，能够恢复丢失分片的各个相邻节点安排其应答分组的发送时间，并且按照序列号连续、低于每 $(1/4)T_r$ 一个分组的速率发送各个应答分组。

3. 主动提取

正如许多 NACK 系统一样，前述提取操作是一种反应式丢失恢复机制，即只有当接收到较大序列号分组时才检测到丢失事件。这种机制会偶尔出现问题，比如，假如丢失一个文件的最后一个分片，那么接收节点无法检测出最后一个分片已经丢失，因为不会再发送较大序列号分组。又比如，假如输入到网络中的文件小，由于突发丢失，常常是丢失所有后续分片，直至包括最后一个分片。假如这样，那么丢失是无法检测的，因此采用反应式丢失检测与恢复机制是无法恢复的。为了解决这些问题，PSFQ 采用基于定时器的主动式提取操作，即假如没有接收到最后一个分片且经过时间 T_{pro} 后不再交付新的分组，那么节点进入主动式提取操作方式，发送 NACK，请求下一个分片或者剩余分片。

主动式提取机制在适当时间自动触发提取方式。假如提取方式触发过早，那么可能浪费额外控制消息，因为上行节点可能在中继消息或者可能还没有接收到必需分片；反之，假如提取方式触发太迟，那么目标节点可能浪费太多时间等待文件最后一个分片，导致文件总交付时延明显增大，正确选择 T_{pro} 必须考虑这两种情况。在重新分配任务应用中，为了便于重新分配任务操作，必须将文件的每个分片保存在数据存储器或者外部存储器中，假如没有接收到最后一个分片且经过时间 T_{pro} 后没有新分组传递到达，那么主动提取机制发送 NACK，请求所有剩余分片（包括最后一个分片）。 T_{pro} 应该正比于最近所收分组最大序列号 S_{last} 与文件最大序列号 S_{max} 之差（该差值等于该文件的剩余分片数量），即 $T_{pro} = \alpha(S_{max} - S_{last})$ ， $\alpha \geq 1$ 。 α 是一个扩展因子，用于调整触发主动提取机制的时延，在大多数情况下将 α 设为 1。

T_{pro} 的定义保证节点等待足够长时间，直到所有上行节点接收到所有分片为止，然后才进入主动式提取方式。当接近文件末尾时，节点可以提前启动主动式提取操作；当文件还有许多分片没有传输完毕时，节点等待较长时间。这种方法能够很好地适应无线环境的质量。假如信道质量好，那么出现连续分组丢失的可能性小，因此在预定的最后一个分片之前没有接收到新消息的原因很可能是丢失最后一个分片，此时最好立即启动主动式提取操作。反之，当信道误码率高时，节点很可能遇到连续分组丢失，因此在重新发送控制消息之前应该等待较长时间。假如知道 WSN 布置在恶劣的无线环境中，那么 α 应该大于 1，节点在启动主动式提取操作之前等待较长的时间。

在小容量数据存储器的其他 WSN 应用中，节点只能保存其已收分片的一部分，主动提取机制发送 NACK，请求数据存储器所能维护数量（或者更少）的分片量。在这种情况下， T_{pro} 应该正比于数据存储器缓存量。假如数据存储器缓存 n 个分片，那么 $T_{pro} = \alpha n T_{max}$ ， $\alpha \geq 1$ 。

α 的设置方法同前，即在低误码率环境中设为 1，而在恶劣无线环境中设为大于 1 的值。这种方法保持分组序列号间断间隔小于 n ，即确保节点连续丢失 n 个分片后启动主动式提取操作。在分发操作中，节点在中继一条消息前最多等待时间 T_{\max} ，所以从上行节点的数据存储器中寻找到丢失分片的概率达到最大。

主动式提取操作确保所有预定接收节点最终会接收到全部数据。和停止重试之前重试最大次数的任何协议一样，PSFQ 主动式提取机制在达到重试门限值之后也能够停止，但是这是由具体应用来选择的。

4. 基于信号强度的提取

在采用低功率电台、没有频率分集的 WSN 中，分组交付性能随着空间和时间的不同而存在极大变化。断断续续地接收多跳远节点的分组（不论信号多弱）可能引起节点发送不必要的 NACK 和进行多余重传，因此 PSFQ 在提取操作和恢复操作期间还考虑了分组的接收信号强度。节点维护一张父节点表（即接收这些节点发送来的消息），同时记录有关平均信号质量。一个节点检测到所收分组序列号不连续时，假如该分组是一个平均信号质量最好的父节点发送来的，那么该节点只做出响应，发送一个 NACK。这就有效抑制了接收到多跳远节点的分组而触发的不必要 NACK。

类似地，节点发送的 NACK 包含其平均信号质量最好的首选父节点。节点接收到该 NACK 后，确定自己是首选父节点，还是首选相邻节点。所有非首选相邻节点将其发送恢复分组的响应时间加倍，因而具有较大机会旁听到更佳候选节点（即首选的父节点/相邻节点）发送的恢复分组，节点只要在发送恢复分组之前旁听到恢复分组就可以取消其恢复分组的发送。当节点没有好机会将丢失分片交付给提取节点时，这种方法能够有效地防止节点进行多余重传。

9.2.4 PSFQ报告操作

PSFQ 支持经专门设计的可选报告操作。可选报告机制采用简单且可扩展方式将数据交付状态反馈给用户。在无线通信中，发送一个长分组的通信开销低于采用多个较短分组发送同样多数据的通信开销。假如网络中可能存在长路径（即多跳转发长路径提高了数据交付开销）和大量目标节点，又假如每个节点均采用报告消息方式发送反馈，那么网络将不堪重负。因此，有必要使反馈使用的消息最少。

假如一个节点接收到一个数据分组，并且该数据分组头的“报告比特”被置位，那么该节点进入报告方式。用户节点只要需要知道其周围节点的最新状态，则在其发送到网络中的数据分组头中设置报告比特。为了减少报告消息数量、避免发生报告消息暴问题，只是最后一个转发跳节点（即 $TTL=1$ ）立即做出响应，在随机时刻（在 $0\sim\Delta$ 之间）给其父节点发送报告消息（已接收到该父节点发送来的前一个数据分片）。到达源节点的路径上的每个节点将其自己的状态信息添加到报告消息中，然后将累积的报告消息发送给用户节点；若报告消息中已包含自己的 ID，则不予理睬所收报告消息，避免出现闭环。处于报告方式下但又不是最后一个转发跳的节点等待时间 $T_{\text{report}}=T_{\max}\times TTL+\Delta$ ，足够接收最后一个转发跳节点的报告消息，使其携带自己的状态信息；经过 T_{report} 后若是没有接收到报告消息，则初始化自己的报告消息，并将其发送给自己的父节点。假如网络规模非常大，那么有可能节点接收到报告消

息后，由于分组太大而不能再添自己的状态信息，此时节点产生新的报告消息并首先将其发送给父节点，然后再转发所收报告消息，从而确保到达用户节点路径上的其他节点使用较新的报告消息而不会产生新的报告消息（这是因为这些节点也会接收到不能再添加自己状态信息的长报告消息）。

9.2.5 单个分组消息的交付

需要在 WSN 中支持单个微小消息的可靠交付，例如，支持传感器的可靠控制与管理。对于适合单个分组交付的消息（如小于网络 MTU），采用 PSFQ NACK 协议而不采用直接信令不能检测交付失败问题，这是因为 PSFQ 通过观测序列号不连续或者超时来检测分组丢失问题。为此，PSFQ 利用报告原型在中心节点获取特定应用的反馈信息。PSFQ 在中心节点将每条单分组消息（要求可靠交付）中的报告比特置位；中心节点根据反馈状态重传该分组，直到所有接收节点确认成功接收为止。由此驱使 PSFQ 运行中心节点按需使用的完全累积 ACK 协议，处理这些特殊情形消息。使用报告机制支持单个微小消息的可靠交付突出了应用 PSFQ 机制满足特定应用需求的灵活性。

9.2.6 PSFQ 的性能

利用 ns-2 仿真研究 PSFQ 性能，以及讨论 PSFQ 设计选择的益处。仿真结果指出：即使在高误码条件下，PSFQ 也仍然能够在 WSN 中可靠交付数据。

1. 仿真方法

在 ns-2 网络仿真器中，针对重新分配任务应用，实现 PSFQ。为了突出 PSFQ 的各种设计选择，比较 PSFQ、SRM^[7]理想版。SRM 有些类似于 PSFQ 的性质，提供 IP 网络的可靠多目标服务。SRM 支持 IP 层上面的可靠多目标传输，采用三条控制消息进行可靠交付：会晤消息、请求消息和恢复消息。SRM 位于 IP 层上面，所以假定存在一条路径从源节点到达每个接收节点，每个节点对每个多目标分组每次最多接收一次。SRM 也适用于这种网络拓扑：路由器不是组活动成员，没有维护状态，但是需要多目标路由。SRM 代表一种方法，即采用直接信令进行可靠数据交付，而 PSFQ 是最低限度要求更低的传输协议，支持单目标（在路由层上面）和广播传输，不要求周期性信令。

PSFQ 包含丢失检测/恢复机制，而用理想全能多目标路由替代 SRM 中的 IP 多目标传输子层。只比较 PSFQ 与 SRM 的可靠交付性能。因为 PSFQ 使用简单广播机制作为重新分配任务应用的路由机制，所以在仿真中将 SRM 层安排在理想全能多目标路由层上面是合理的。源节点采用全能多目标路由沿着最短路径多目标树将其数据发送给所有预定接收节点，计算最短路径和构建到达每个目的节点的路由树均没有通信开销。

比较的主要目的是突出 PSFQ 各个机制的影响。SRM 是传统的基于接收节点的可靠传输协议的代表，其互联网应用可扩展性强。SRM 服务模型非常类似于 WSN 重新分配任务应用，但是 SRM 是为有线互联网设计的，有线互联网的传输媒介可靠性高、不存在 WSN 中的那些独特问题（如隐含终端、干扰等）。为了公平比较，将低层理想化，使传输媒介的差异降到最低程度，并且将仿真重点只放在可靠数据交付机制上，将这种理想化的 SRM 称为 SRM-I。

采用如下三个性能指标进行对比评估：

① 平均交付率：表示目标节点成功接收到的消息数量与用户节点发送到网络中的消息数量之比率。平均交付率说明传输协议在一定时间限制内交付率低于 100% 时的容错能力。

② 平均时延：表示用户节点第一个数据分组发送时刻与 WSN 中最后一个目标节点成功接收到用户节点最后一个分组时刻之间的平均时间。平均时延说明传输协议的时延范围。

③ 平均交付开销：表示一个目标节点每次成功接收到一条数据消息时所发送的总消息数量。平均交付开销说明在网络上实现可靠交付的通信开销。

通过仿真研究这三个性能指标与信道误码率、网络规模之间的变化关系。

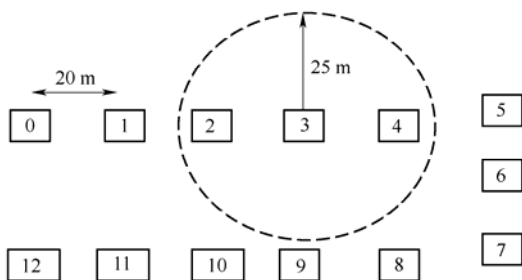
为了评估 PSFQ 在实际情况下的性能，评估 PSFQ 在灾后重建 WSN 的重新分配任务应用中的性能：沿着建筑物各个地面走廊布置传感器节点。图 9-3 (a) 给出一个简单的 WSN：网络分布在 100 m×100 m 的区域内；每个传感器节点相互距离 20 m，采用 2 Mb/s 传输速率和标称无线传输距离 25 m 的电台；信道访问协议为 CSMA/CA；采用均匀分布信道误码模型；位于位置 0 上的用户节点给每个传感器节点发送 2.5 KB 程序映像文件，以便重新分配每个传感器节点的任务；分组大小为 50 B；以每 10 ms 一个分组的速率发送用户节点产生的分组；对于 PSFQ，保守设置定时器参数， $T_{\max}=100$ ms， $T_{\min}=50$ ms， $T_r=20$ ms，因此提取操作比分发操作快 5 倍。每个实验做 10 次，取 10 次实验结果的平均值。

2. 仿真结果

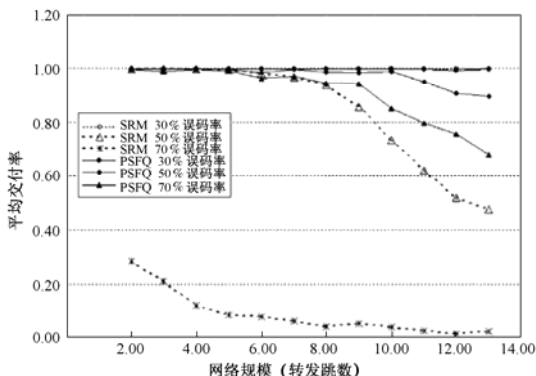
PSFQ 的主要目标之一是保证在各种无线信道条件下能够正常工作。第一个仿真实验是检验 PSFQ 和 SRM-I 的“容错”实验，比较其实验结果。

图 9-3 (b) 给出了 PSFQ 和 SRM-I 在各种信道误码率条件下的容错性能与网络规模之间的变化曲线。正如预期的那样，PSFQ 和 SRM-I 的平均交付率随着信道误码率的提高而下降。在较高误码率的情况下，当网络规模增大时，交付率迅速下降。用户节点在仿真时间达到 2 s 时开始以每 10 ms 一个分组的恒定速率给网络发送数据分组，在 0.5 s 时间内完成全部 50 个分组的发送。用户节点完成数据分组发送后继续仿真 100 s。从图 9-3 (b) 中看到：SRM-I (虚线) 只有在信道误码率低于 30% 的时候才能够实现 13 个转发跳内的 100% 数据交付率；在 50% 的信道误码率下，SRM-I 只能实现 5 个转发跳内的 100% 数据交付率；对于更高的信道误码率，SRM-I 只能在 2 个转发跳内完成文件部分数据的可靠交付。再观察图 9-3 (b) 中各种信道误码率下的 PSFQ (实线) 交付率均优于 SRM-I：在 50% 的信道误码率下，PSFQ 也能够实现 10 个转发跳内的 100% 数据交付率，实现 13 个转发跳内高于 90% 的交付率；即使在 70% 的严重信道误码率下，PSFQ 仍然能够实现 4 个转发跳内的 100% 数据交付率，实现 13 个转发跳内高于 70% 的交付率，而 SRM-I 只能实现 2 个转发跳内低于 30% 的交付率。

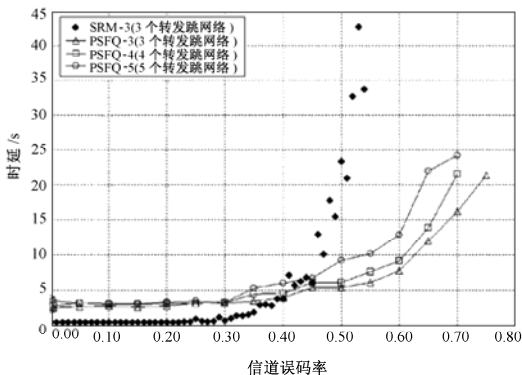
PSFQ 的容错能力优于 SRM-I 证明 WSN 慢分发快提取设计思想是正确的。连续序列号数据分发操作能够防止丢失事件的传播。SRM-I 没有提供按序数据交付机制，丢失事件会沿着多目标树传播。PSFQ 的主动提取操作和丢失累积技术支持单条控制消息包含多个丢失窗口。SRM 依靠低层 (MAC 层协议) 在多目标组成员节点间可靠交付直接的周期性控制消息。IEEE 802.11 MAC 中的虚拟载波侦听在高误码率环境中的失效导致 SRM-I 失效，而 PSFQ 对底层的最低限度要求使其即使在高丢失率条件下也能够有效控制广播。



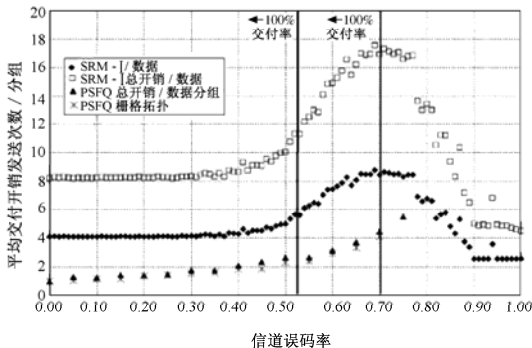
(a) 布置在建筑区内的 WSN(布置在位置 0 上的一个用户节点在 0.5s 内给网络发送 50 个分组)



(b) PSFQ 与 SRM-I 的容错比较 (在各种信道误码率下平均交付率与转发跳数之间的变化关系)



(c) PSFQ 与 SRM-I 的交付时延比较



(d) PSFQ 和 SRM-I 的平均交付开销比较

图 9-3 PSFQ 与 SRM-I 的对比

第二个仿真实验是 PSFQ 和 SRM-I 在各种信道误码率下的数据交付时延实验, 实验结果如图 9-3 (c) 所示。只有当所有预定目标节点在仿真结束前接收到所有数据分组的时候才能够确定交付时延。当信道误码率高时, SRM-I 只能实现有限几个转发跳内的 100% 交付率。在本实验中, 使用 3 个转发跳网络比较 PSFQ 和 SRM-I 的交付时延, 而只针对 PSFQ 研究其较大转发跳数内的交付时延 (因为 PSFQ 的容错能力强于 SRM-I)。从图 9-3 (c) 中看到: 当信道误码率低于 40% 时, SRM-I 的交付时延小于 PSFQ, 但是 SRM-I 的交付时延随着信道误码率的增大而呈指数增大, 而 PSFQ 的交付时延在信道误码率低于 70% 时较缓慢地增大。原因在于 PSFQ 的“慢分发”机制: 每个节点延迟一段随机时间 (位于 T_{\min} 和 T_{\max} 之间) 后才会转发分组。尽管 PSFQ 在较低信道误码率下的交付时延差于 SRM-I, 但是将“慢分发”机制与“快提取”机制结合起来使用证明是非常有效的。如图 9-3 (c) 所示, PSFQ 即使在极高信道误码率下也仍然能够保证交付时延。

第三个仿真实验是 PSFQ 和 SRM-I 在各种信道误码率下的可靠交付通信开销实验。采用 3 个转发跳实验网络, 16 个节点布置在 4×4 网络栅格上, 研究 PSFQ 在密集网络 (最多具有 4 个相邻节点) 中的通信开销。按照每个数据分组的平均发送次数来测试通信开销 (即平均交付开销)。对于 SRM-I, 从总通信开销中分离出 SRM 特定的丢失恢复机制的通信开销, 包括有关链路层丢失恢复机制 (RTS/CTS/ACK) 的通信开销。实验结果如图 9-3 (d) 所示。从图 9-3 (d) 中看到: 即使不考虑 SRM-I 的链路层通信开销, PSFQ 的通信开销仍然是低于 SRM-I 一个数量级; PSFQ 在密集栅格网络中的通信开销非相称, 但是低于链式网络拓

扑的通信开销,这说明 PSFQ 能够利用相邻节点冗余度同时抑制不必要的冗余发送。图 9-3 (d) 还说明了 PSFQ 和 SRM-I 的 100% 交付率阻碍点(两条垂直线):52% 信道误码率标记是 SRM-I 的 100% 交付率阻碍点,70% 信道误码率标记是 PSFQ 的 100% 交付率阻碍点。PSFQ 和 SRM-I 的仿真性能差异证明 PSFQ 各种机制是正确的、有效的。PSFQ 利用被动、按需丢失恢复机制,SRM 利用周期性交换会晤消息的丢失恢复机制。

9.3 下行数据可靠交付可扩展体系结构 (GARUDA)

WSN 迫切需要(因此缺乏)可靠性显然与其特定应用密切相关。考虑安全应用,要求图像传感器检测和识别关键目标的出现与存在。若已知安全应用的主要特征,则从中心节点发出的所有消息必须安全到达传感器。

对于安全应用,中心节点可能发送如下三类消息:①假如低层网络由可配置传感器组成,那么中心节点可能需要给传感器发送特定的(比如升级版)图像检测/处理软件,将这种消息称为控制消息。②中心节点可能必须给传感器发送一个目标图像数据库,帮助随后查询触发的图像识别,将这种数据称为查询数据。③中心节点可能发送一个或者多个查询,申请有关某个特定目标的检测信息;传感器接收到中心节点的查询后,将其与已存储的图像比较,根据比较结果做出相应的响应。

无线 Ad Hoc 网络中的可靠数据交付技术不能直接应用到传感器环境中,理由如下:①环境考虑。WSN 环境带来的约束极不同于其他多跳无线网络带来的约束,比如网络节点寿命有限、带宽和能量的稀缺、网络规模等。②消息考虑。尽管在多跳无线网络中大多数节点组可靠传输方法考虑长消息(跨越几个分组长度),但是 WSN 中的大多数消息可能是短小的查询消息,因此需要研究采用哪种类型的丢失恢复体制。③可靠性考虑。传统上流行的可靠性概念是简单的 100% 可靠数据交付。但是,WSN 可能需要其他可靠性概念,从只对网络某个子区域的可靠交付到基于范围解析的查询的局部概率可靠性。

针对上述挑战,美国佐治亚州理工学院提出一种叫做 GARUDA 的解决方法。GARUDA 提供从中心节点到传感器的点到多点的可靠数据交付。就网络规模、消息特征、丢失率、可靠性语义而言,GARUD 是可扩展的。GARUDA 的设计基础如下:①基于高效脉冲的短消息可靠交付方法;②利用虚拟基础设施(称为 GARUDA 核)实现本地指定服务器的近似最佳分配,GARUDA 核是在分组泛洪期间即时构建的;③基于两阶段 NACK 的恢复过程能够有效将重传过程降到最低程度,采用乱序转发,在 WSN 中有力支持最大可能空间复用;④采用简单的候选法支持 WSN 可能要求的不同可靠性概念。

9.3.1 面临的挑战

GARUDA 体系结构是针对 WSN 中中心节点到传感器的下行可靠数据交付问题的,其研究范围严格限制为单个中心节点、固定传感器的 WSN,包括解决 WSN 可能需要的各种可靠性语义问题,同时假定缺乏通信可靠性的原因是各种各样的,比如随机信道误码、拥塞以及其他失效。因此 GARUDA 的研究目标是采用既针对 WSN 独特特点,又具有适当平衡作用的方法实现可靠性,同时使带宽使用率、能耗、时延最低。

下面分析在 WSN 中实现可靠下行交付所面临的基本挑战。

1. 环境限制

为了提供有效的下行数据可靠性，需要处理 WSN 的两个主要约束条件：① 带宽和能量的限制；② 频繁的节点失效。

处理带宽和能量限制时要求重传次数开销最低，以确保可靠性。反之，由于可靠性过程，重传次数开销的降低又降低了带宽的使用和能量的消耗。不能采用固定结构化机制（如广播树）处理节点易失效问题，因为固定结构化机制没有考虑网络的动态性。由于可靠性过程的开销必须最低化，所以不需要周期性刷新结构的动态机制。

必须考虑的另一个目标环境特征是网络扩展性。WSN 可能由大量传感器节点组成，因此网络直径大。这就意味着网络中可能存在极大的空间复用和时延，因此应该利用空间复用实现最大容量。但是，具体采用的丢失恢复机制可能严重限制空间复用。

2. ACK/NACK

尽管环境挑战是关于环境带来的各种约束，但是 ACK/NACK 挑战是下行可靠性可能采用的典型消息类型带来的约束。尽管查询数据和控制代码不是短消息，但是查询消息是短消息，因此引起一个独特的问题。

只要丢失率不是过高，NACK 通常在组通信，尤其是在多跳无线网络中作为一种有效处理丢失广播机制。但是，NACK 不能处理网络中某个特定节点丢失一条消息中所有分组这种情况。这是因为节点不知道消息何时到达，所以节点不能广播 NACK 而要求重传。

假如是长消息，那么该消息中所有分组没有传递到达某个节点的概率忽略不计。但是，对于查询之类的短消息，认为消息只包含少数几个分组是非常合理的，则不能忽视一条消息中所有分组没有传递到达某个节点的概率，因此必须明确加以处理。

尽管基于 ACK 的恢复方法能够解决丢失问题，但是由于 ACK 的其他缺点（ACK 暴），因此显然不能使用 ACK。

空间复用、基于 NACK 的丢失恢复方法需要网络节点按序转发数据，以防 NACK 暴。这显然限制了网络中的空间复用。

9.3.2 可靠性语义

WSN 环境固有的两个特征是布置传感器时的位置依赖性和冗余度。查询可以与位置有关，比如“发送房间 X、Y、Z 的温度”。在感知场中布置冗余传感器意味着：为了获得可靠感知信息，没有必要感知场中所有传感器将其本地感知数据可靠交付给中心节点。中心节点也可以随机选择网络某个部分进行消息可靠交付，比如递增解析感知策略。

因此，根据上述特征定义 WSN 需要的可靠性语义。将可靠性语义分成 4 类：①对整个传感器场的可靠交付（默认语义）；②对传感器场某个子区域中传感器的可靠交付，这是基于位置交付的典型代表；③对覆盖整个传感器场的传感器的可靠交付，这是冗余意识交付的典型代表；④对一个随机传感器子集的可靠交付，这与范围解析应用一致。图 9-4 说明了这四种类型的可靠性语义。任何一种可靠性解决方案不仅应该支持默认可靠性语义，而且还应该支持无线传感器环境所独有的其他类型可靠性语义。

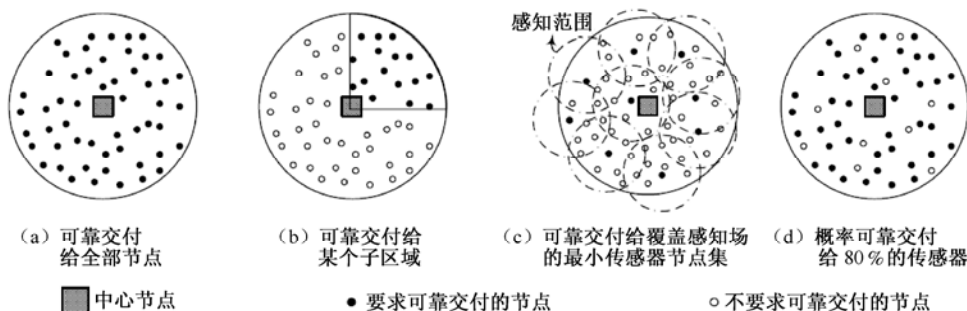


图 9-4 可靠性语义分类

9.3.3 GARUDA的基本原理

GARUDA 的主要特征是即时可构建丢失恢复基础设施（称为 GARUDA 核）。GARUDA 核近似于网络图的最小支配集（Minimum Dominating Set, MDS），要求其可靠交付消息。尽管使用 MDS 概念来解决网络问题不是新技术^[23]，但是针对特定目标环境在 GARUDA 中运用 MDS 的作用如下：不论 GARUDA 核如何构建、如何用于丢失恢复以及如何支持多种可靠性语义，总是对丢失恢复过程 GARUDA 核实现相对最优化。

GARUDA 的基础是脉冲法，用于将单个分组可靠交付给所有网络节点。因为 GARUDA 能够确保可靠交付任意长度消息的第一个分组，所以 GARUDA 不再像直接转发 NACK 那样面对所有分组丢失问题时显得束手无策。这就使得 GARUDA 既能利用 NACK 法的优点，又能避免采用 NACK 法带来的缺陷。

1. 丢失恢复服务器：GARUDA核

GARUDA 在其丢失恢复过程中采用本地指定丢失恢复服务器。采取本地恢复是为了减少非本地服务器的瓶颈，采用指定服务器是为了预防相邻节点接收到重传请求后作出不必要的冗余重传。因此，GARUDA 核形成本地指定丢失恢复服务器集，辅助丢失恢复过程。其间面临的挑战是：① 应该如何选择 GARUDA 核心节点才能够使重传开销最低？② 按照一种适合于目标环境约束特征（节点失效引起的动态拓扑变化）的方法如何构建 GARUDA 核？

在理想情况下，应该根据每个分组以及在分组交付期间所经历的丢失模式指定 GARUDA 核。一旦已知丢失模式，那么指定最佳服务器（根据所要求的重传次数）就简化为最小集覆盖问题（Minimum Set-Cover, MSC）^[19]。尽管 MSC 问题的解决方法是理想方法，但是就根据每个分组指定核而论，显然是一个不可行方法。

GARUDA 根据每条消息指定其核，与分组丢失模式无关。以较大时间尺度指定丢失恢复服务器不利于支持网络动态性和不同可靠性语义。GARUDA 指定丢失恢复服务器的方法是：为每条消息交付从网络中动态选择一个节点子集。尽管 GARUDA 核对于每种分组丢失模式不是最佳的，即 GARUDA 核不是特定丢失模式的最小集覆盖范围的近似，但是却近似于最小支配集。若假定传感器节点均匀布置，那么可以证明：对于任意丢失模式，若构建 GARUDA 核时使用近似最小支配集，那么最差情形比率 $[1 + \ln(n)]/[1 + \ln(n) + \ln(p)]$ 约等于

MSC 结构的多项式时间近似, p 表示丢失概率 ($0 \leq p \leq 1$), n 表示网络节点总数。

运用第一个分组交付构建 GARUDA 核。第一个分组的可靠交付决定节点与中心节点之间的转发跳距离 (hop_count)。若是一个节点的 hop_count 等于 3 的整数倍, 并且假如该节点没有旁听到任何其他 GARUDA 核心节点的发送, 那么该节点选择自己作为 GARUDA 核心节点。这种核心节点选择规程类似于分布式 MDS 结构。GARUDA 核具有如下独特性: ①GARUDA 核是运用单个分组泛洪 (特别是在第一个分组泛洪期间) 来建立的; ②平衡 WSN 拓扑结构 (传感器与中心节点之间的距离固定不变), 以获得更加高效而公平的 GARUD 核结构。在每条消息的第一个分组交付期间即时构建 GARUDA 核有助于改善网络在面对节点失效 (在消息间隔发生) 问题时的脆弱性。

2. 丢失恢复过程

(1) 采用可用性位图传播的乱序转发

在 GARUDA 中, 不是采用按序策略而是采用乱序策略转发分组。按序转发策略的主要缺点是: 在发生丢失事件后, 当较大序列号分组被禁止转发时, 保留的下行网络资源未被充分利用。而乱序转发策略克服了这个缺点, 因为丢失了分组的节点可以继续转发所接收到的较大 (或者较小) 序列号分组。但是, 乱序转发策略可能会产生不必要的 NACK 暴。下行节点发出一系列 NACK 请求丢失分组, 即使当有关分组不再可用也是如此。

为了抑制不必要的重传, GARUDA 采用可扩展的可用性位图 (Availability Map, A-map) 交换法: 在 GARUDA 核心节点之间交换 A-map, A-map 包含元级信息, 通过比特置位来表示分组的可用性。下行核心节点只有在接收到其上行核心节点发送的 A-map, 并且其中相应比特被置位的时候才会开始请求丢失分组。核心节点只有在确信上行核心节点有丢失分组的时候才会请求该丢失分组, 因此 GARUDA 核恢复阶段非常高效。显然需要考虑 A-map 带来的开销, 后面介绍的 GARUDA 仿真性能结果考虑了 A-map 开销, 因此所给出的任何性能改善都考虑了 A-map 开销。

(2) 两阶段丢失恢复

一旦完成 GARUDA 核的建立, GARUDA 就立即进入两阶段恢复过程: 首先 (第一阶段) 是核心节点恢复所有丢失分组, 然后 (第二阶段) 是非核心节点恢复丢失分组。GARUDA 将恢复过程分成两个阶段进行的主要理由是: ① 非核心节点在网络节点总数中占相当大一部分, 因此需要排除与非核心节点的竞争; ② 当核心节点重传丢失分组给其他核心节点时, 应该使用单个重传在核心节点的相邻节点之间将对应于单个分组的序列号漏洞填满; ③在第二阶段只有核心节点在执行重传时, 由于 GARUDA 核的特性 (在理想情况下, 相距两跳范围内不存在两个核心节点), 不同核心节点的重传发生碰撞的概率达到最低。

① 核心节点的丢失恢复: 核心节点丢失恢复过程与基本默认消息转发按并行方式同时执行, 其目的是为了确保核心节点尽可能快地接收到消息中的全部分组。由于核心节点只占网络节点总数中极少部分, 所有请求和重传均采用单目标方式传输给最近上行核心节点 (有丢失分组复制), 所以核心节点并行恢复过程不会明显提高网络竞争程度。

② 非核心节点的丢失恢复: 非核心节点只有在旁听到核心节点发送的 A-map, 并且 A-map 说明该核心节点已经接收到消息中的全部分组的时候才会启动丢失恢复的第二阶段。因此在每个本地区域中, 丢失恢复第二阶段不会与丢失恢复第一阶段重叠, 从而防止了与基本泛洪机制以及第一阶段恢复的竞争。

GARUDA 的两个阶段恢复可能会引起时延的增大，但是相对于竞争方法，这种时延的增大是非常小的。

3. 多种可靠性语义

对上述 GARUDA 核结构进行简单修改，就可支持前述多种可靠性语义。不失一般性，首先假定一个可靠性语义实例要求对基本图 G 中的节点子图 G_S 进行可靠交付；子图 G_S 由 K 个部分组成，每个部分都是连通的，但是各个部分不是相互连通的。假如优化准则是带宽开销^[25]，那么所需基础设施是必须计算每个部分的 MDS，并且利用旅行推销员路径 (Travelling Salesman Path, TSP) 将各个部分连接到中心节点。

GARUDA 使用一种较简单且又相当有效的技术计算各个 MDS，然后利用近似最短路径树 (Shortest Path Tree, SPT) 将各个 MDS 连接到中心节点。这种方法存在额外的带宽开销，但是改善了时延性能，而且其可重构性是固有的。图 9-5 (a) 说明了 GARUDA 解决方法：求每个部分的 MDS，将所有 MDS 连接到中心节点的近似 SPT。

用来构建每个部分 MDS 的 GARUDA 核结构算法稍修改如下：节点在参与 GARUDA 核结构算法前需要进行候选资格检查，只有通过资格检查的方可参加。候选资格检查就是接收到第一个分组的节点确定是否属于子图 G_S 。 G_S 外的节点若是要求构建 SPT，则通过强迫候选资格机制而被包含在 GARUDA 核结构中。

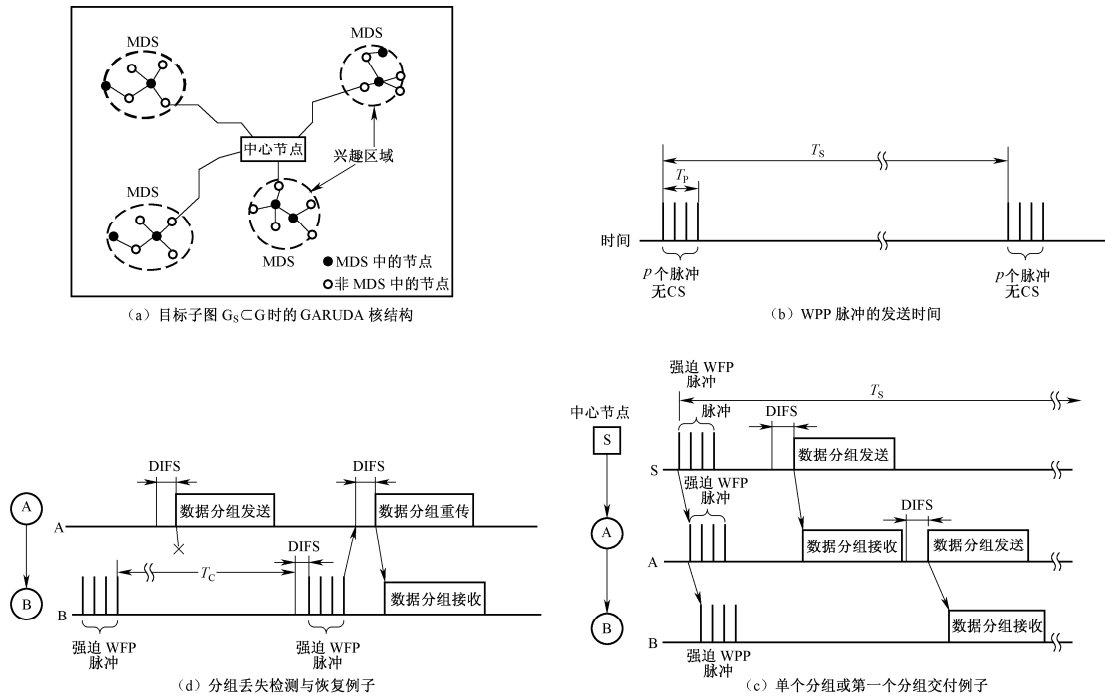


图 9-5 GARUDA 核结构与单个分组/第一个分组的交付

4. 单个分组或第一个分组的可靠交付

迄今为止，已经在第一个分组可靠交付给所有网络节点的假设条件下详细讨论了

GARUDA 核基础设施。下面概述如何实现第一个分组的可靠交付。

NACK 请求法在没有任何支持下不能满足单个分组交付的要求（或者当丢失消息中全部分组之时），所以考虑将 **ACK** 法作为第一个分组可靠交付的一种备用方法。只要将一个具有足够多有关消息的信息（如长度）的分组交付给每个节点，采用 **NACK** 法就能成功提供有保证的可靠性。但是，这种备用方法仍然存在 **ACK** 暴问题。

GARUDA 采用等待第一个分组（Wait-for-First-Packet, **WFP**）脉冲解决第一个分组可靠交付问题。**WFP** 脉冲是一个有限的短持续时间脉冲序列，并且是周期性重复序列。**WFP** 脉冲幅度比正常发送的数据脉冲幅度大得多。**WFP** 脉冲的独特性是：由于 **WFP** 脉冲幅度/周期的特殊性，任何接收节点，不论当前是空闲还是在接收正常数据分组，都能够侦听到 **WFP** 脉冲。

中心节点需要发送第一个分组时周期性发送 **WFP** 脉冲有限序列。中心节点覆盖范围内的传感器节点接收到 **WFP** 脉冲后，以相同周期在两个 **WFP** 脉冲序列之间发送 **WFP** 脉冲，反复重复这个过程，直到所有节点接收到第一个数据分组后开始发送 **WFP** 脉冲为止。中心节点经过一段时间有限长度的 **WFP** 脉冲发送之后（以确保 **WFP** 脉冲在网络中传递通过了多个转发跳），停止发送 **WFP** 脉冲，然后按照正常数据分组发送方式发送第一个数据分组。接收到第一个数据分组的每个传感器也停止发送 **WFP** 脉冲，然后按照正常数据分组发送方式发送第一个数据分组。

WFP 信号实质上起两个作用：①允许中心节点通知传感器其即将发送的消息具有可靠性要求；②当传感器没有成功接收到第一个数据分组时，**WFP** 脉冲能够使传感器请求重传。根据能耗以及 **WFP** 机制的额外开销，资源有限的传感器有可能出现过载，但是采用 **WFP** 信号减轻了消息可靠交付的几个有关问题，实际上得到的益处远胜于其带来的额外开销。

总之，由于 **WFP** 脉冲只是用来表示即将进行的新发送将要到达，所以要求其调制方式比默认数据传输简单，但是抗衰落效应能力更强；采用 **WFP** 脉冲的消息广播方法本身抗碰撞能力强，因为 **WFP** 脉冲与其他 **WFP** 脉冲或者数据发送的碰撞不会妨碍传感器旁听 **WFP** 脉冲，并据其推断即将进行的消息发送（传感器仍然能够侦听到正在发送的 **WFP** 脉冲）；**WFP** 机制不同于 **ACK** 法，**ACK** 暴对数据传输具有不利影响，**ACK** 法的网络规模可扩展性差，而 **WFP** 脉冲作为一种间接 **NACK**，**WFP** 脉冲带宽窄，因而 **WFP** 脉冲对正常数据传输的干扰达到最低程度；**WFP** 脉冲的能耗明显低于正常数据传输的能耗，因此由于得到其他益处，**WFP** 脉冲带来的额外能耗远小于实际节能。

下面针对一个简单基本泛洪机制以及在全网节点 100%交付语义下介绍 **GARUDA** 体系结构。但是，**GARUDA** 本身能够很好地与这个泛洪机制综合在一起。假定输入的每个泛洪分组是具有可靠性要求的消息的组成部分，则将其交付给 **GARUDA**。

按照泛洪可靠消息时的时间发生顺序描述 **GARUDA** 体系结构的各个组成部分。因此，首先详细描述 **GARUDA** 基于脉冲的单个分组交付机制；然后描述 **GARUDA** 核的结构和丢失恢复规程；最后介绍 **GARUDA** 对多种可靠性语义的支持。

9.3.4 单个分组或第一个分组的交付

1. 发送WFP脉冲

WFP 脉冲是短周期信号，不包含任何信息，所以 **WFP** 脉冲的发送周期明显小于正常数

据分组所需要的发送时间 T_D 。WFP 脉冲的发送功率是正常数据分组发送功率的 2 倍，以便在接收机达到 3 dB 的相对幅度（默认接收）。接收机根据能量检测策略（监视输入信号能量幅度的变化以及变化持续的时间）检测 WFP 脉冲。即使接收机信道正在忙于数据发送，接收机也仍然能够检测出能量变化。只有没有旁听到 WFP 脉冲的节点才是不在侦听的节点（要么在发送方式，要么在关机方式）。

为了提高 WFP 脉冲检测的强壮性，每个 WFP 脉冲集包含 p 个脉冲，在时间周期 T_P ($T_P \leq T_D$) 内连续发送完这 p 个脉冲。图 9-5 (b) 表示 WFP 脉冲的发送方法。因此，接收节点只是在检测到 p 个脉冲之后才推断 WFP 输入信号。

GARUDA 的基本 WFP 脉冲机制（唯一要求）没有采用载波侦听，因此被称为强迫 WFP 脉冲。这种机制确保需要发送 WFP（或者广播、或者请求第一个分组的 NACK）的节点能够发送而不会遇到 MAC 层饥饿问题。但是，这种发送显然会提高与正常数据分组发送的碰撞概率，因此按照周期 T_S ($T_S \geq T_D$) 进行发送。

但是，GARUDA 强迫 WFP 脉冲优于基于载波侦听的 WFP 以及基于数据分组携带的广播法，后两者减轻了强迫 WFP 脉冲的影响。但是，只采用强迫 WFP 脉冲保证第一个分组可靠交付。

2. GARUDA 的第一个分组交付

单个分组或第一个分组的交付规程由三种方式组成：①广播方式——广播通知采用强迫 WFP 脉冲的所有节点即将发送单个/第一个分组；②交付方式——通过简单转发发送单个/第一个分组；③恢复方式——采用 WFP 脉冲发送 NACK，请求重传单个/第一个分组。

图 9-5 (c) 以一个简单拓扑为例给出单个分组或者第一个分组的基本交付规程。中心节点需要启动单个分组或第一个分组的可靠交付时，发送一系列强迫 WFP 脉冲（不进行信道侦听）。相邻传感器接收到 WFP 脉冲后，立即发送一系列强迫 WFP 脉冲。经过一段时间（根据网络直径来设置）后，中心节点根据 MAC 协议（如 CSMA）发送单个/第一个数据分组。

节点 A 接收到单个/第一个分组后，从广播方式切换到交付方式，停止发送 WFP 脉冲，经过载波侦听后发送单个/第一个分组。但是，若是丢失单个/第一个分组，那么节点继续发送 WFP 脉冲，由此触发重传单个/第一个分组。图 9-5 (d) 给出了一个丢失检测与恢复例子。

由于按周期 T_S 发送的强迫 WFP 脉冲起 NACK 信号作用，所以节点 B 至少等待时间 T_S ，然后才发送下一个强迫 WFP 脉冲系列。因此，单个/第一个分组的交付时延直接依赖 T_S 。

为了降低单个/第一个分组的交付时延，GARUDA 采用另一种 WFP 脉冲机制，即节点经过正常载波侦听后才发送单个/第一个分组。节点 B 经过正常载波侦听 after 按周期 T_C ($T_C < T_S$) 概率发送 p 个 WFP 脉冲 (WFP_{CS})（除非已经接收到单个/第一个分组）。周期 T_C 应该正比于节点 B 与中心节点之间的转发跳距离，这是因为节点应该等到其与中心节点之间的上行节点接收到单个/第一个分组为止。 T_C 设置如下

$$T_C = i \times \Delta \times T_D, \quad (9-2)$$

式中， i 表示从中心节点到达一个传感器节点的转发跳距离， Δ 表示最大节点密度。

由于节点在发送 WFP_{CS} 脉冲前侦听信道状态，所以 WFP_{CS} 脉冲与数据分组的碰撞概率低于 WFP 脉冲与数据分组的碰撞概率。一个节点准备发送 WFP_{CS} 脉冲时，将定时器设为强迫 WFP 脉冲的时间周期 T_S 。

GARUDA 采用的概率优化就是让正常数据分组携带 NACK 信息。NACK 只不过是节点

迄今已接收到的最新消息 ID 的序列号。任何相邻节点获悉较大消息 ID，且有相应的第一个数据分组，则重传该数据分组。这就是间接 NACK 机制。

9.3.5 即时构建GARUDA核

1. GARUDA核

假定 100%全网范围内可靠泛洪，下面将详细描述即时构建 GARUDA 结构。

假定中心节点位于传感器场中心位置，根据节点到达中心节点的转发跳距离，通过第一个分组交付设置节点组带号 (band-id)，如图 9-6 (a) 所示。具有相同 band-id 的所有节点形成一个具有特定 ID 的“组带”。因此，可以将各个组带看成以中心节点为中心的同心圆。

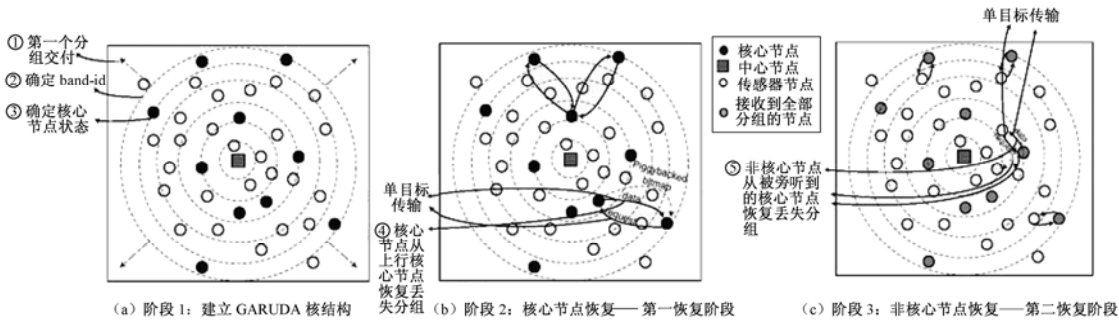


图 9-6 即时构建 GARUDA 核结构以及 GARUDA 丢失恢复

2. 构建规程

GARUDA 核结构算法工作原理如下：

(1) 中心节点

中心节点发送第一个分组时，在第一个分组上加上等于零的组带号。为了平衡核心节点和非核心节点的载荷，中心节点可以选择 band-id 等于 0、1、2。因此 $3i$ 组带（核心节点组带）可能是动态变化的。一个传感器成功接收到第一个分组时，将其 band-id 加 1，将新得到的 band-id 作为自己的 band-id。这个 band-id 表示从中心节点到达该节点的近似转发跳个数。由于从中心节点到达传感器存在多条可用路径，所以计算得到的 band-id 有可能大于从中心节点到达传感器的最小转发跳数。

(2) $3i$ 组带上的节点

① 只允许 band-id 等于 $3i$ (i 是正整数) 的传感器选择自己作为核心节点。

② 一个 $3i$ 组带节点 S_0 接收到第一个数据分组，然后等待一段随机时间，其间若是没有旁听到相同组带内任何其他核心节点，那么在转发第一个数据分组时选择自己作为一个核心节点。一旦一个节点选择自己作为核心节点后，那么发送的所有分组（包括第一个分组）携带相同 band-id 信息。

③ 核心组带内没有选择自己作为核心节点的任意节点若是接收到核心请求消息，则在本阶段选择自己作为核心节点。

④ $3(i+1)$ 组带内的每个核心节点 S_3 应该至少知道 $3i$ 组带内的一个核心节点，若是通过

$3i$ 组带内的一个核心节点接收到第一个数据分组，那么就间接知道了 $3i$ 组带内的这个核心节点，因为每个分组携带先前已被访问的核心节点的 ID 以及 A-map。为了处理 $3(i+1)$ 组带内的一个核心节点 S_3 不知道 $3i$ 组带内任何核心节点这个问题， S_3 维护有关接收到第一个数据分组的节点 (S_2) 的信息， S_2 维护有关接收到第一个数据分组的节点 (S_1) 的信息。经过一段时间 (核心节点选择定时时间) 后， S_3 直接给 S_2 发送一条上行核心节点请求消息，而 S_2 又将该消息转发给 S_1 。至此， S_1 已经选择一个核心节点，因此使用相关信息做出响应。

(3) $3i+1$ 组带上的节点

band-id 等于 $3i+1$ 的一个传感器节点 S_1 接收到第一个分组后，检查该分组是从核心节点还是非核心节点发送来的。假如源节点 S_0 是一个核心节点，那么 S_1 将其核心节点设为 S_0 ；否则，即源节点 S_0 不是核心节点，那么 S_1 将 S_0 设为候选核心节点，启动核心节点选择定时器 (定时时间大于第一个数据分组重传时间)。假如 S_1 在定时结束前旁听到一个核心节点 S'_0 ，则将其核心节点设为 S'_0 ；假如 S_1 在定时结束前没有旁听到任何其他核心节点，则将其核心节点设为 S_0 ，然后按照单目标方式将这个决策通知 S_0 。

(4) $3i+2$ 组带上的节点

band-id 等于 $3i+2$ 的一个传感器节点 S_2 接收到第一个数据分组时，(在此位置) 不知道任何 $3(i+1)$ 传感器，因此转发第一个数据分组而不能选择其核心节点，启动核心节点选择定时器。 S_2 若是在定时结束前旁听到 $3(i+1)$ 组带内一个核心节点，则选择该核心节点作为自己的核心节点；否则，选择所旁听到的 $3(i+1)$ 组带内任意一个传感器节点作为自己的核心节点，然后通过单目标方式将这个决策通知给该传感器节点。 S_2 若是没有旁听到 $3(i+1)$ 组带内任何节点 (有可能发生，但是极少发生)，则采用任意目标方式发送一条核心节点请求消息，该消息只包含取值等于 $3(i+1)$ 的目标 band-id。允许 $3(i+1)$ 组带内任何节点接收到该消息后经过一段随机等待时延后做出响应，随机时延较小，以便于核心节点重复使用已经选定的核心节点。

一个 $3i+2$ 组带节点正好位于网络边沿时出现边界条件，使 $3i+2$ 组带作为候选核心节点组带 ($3i$)。 $3i+2$ 组带内节点没有接收到任意组播核心节点请求消息的响应，则说明检测到这种边界条件。

因此，在第一个分组交付阶段结束之时，每个节点不论是否为核心节点均知道其 band-id，假如不是核心节点则还知道其核心节点信息。此外， $3(i+1)$ 组带内每个核心节点至少知道 $3i$ 组带内一个核心节点。

3. GARUDA核的优化

由于 GARUDA 核心节点近似于最小支配集，所以明显存在一个问题是如何使 GARUDA 核结构使用最少的核心节点。理想情况下，对于任一给定核心节点，其两跳相邻区域内不应该存在任何其他核心节点。GARUDA 体系结构采用二分叉法来达到这个条件：①只允许 $3i$ 组带 (核心带) 内的节点竞争成为核心节点；②对于核心带内的节点，只允许没有旁听到其组带内任何其他核心节点的节点选择自己作为核心节点。

9.3.6 两阶段丢失恢复

1. 核心节点丢失的恢复

核心节点接收到一个乱序分组，则推断发生分组丢失。核心节点若是通过 A-map 获知其

上行核心节点有其丢失分组，则给该上行节点发送一个重传请求。

采用 A-map 对于核心节点恢复过程非常重要。为了简单起见，假定 A-map 能够表示一条消息的全部分组且与消息大小无关。核心节点丢失恢复过程工作原理如下：

(1) 上行核心节点

一个核心节点在转发一个分组时，在该分组添加元信息 (C_{id} , A-map, band-id, vFlag)，其中 C_{id} 表示核心节点的身份识别码，A-map 表示位图，band-id 表示组带号，vFlag 表示有效标志。接收核心节点根据有效标志确定该元信息中的传递路径是否有效。

核心节点接收到一个重传请求后，按照单目标方式重传所需分组。

(2) 中间非核心节点

任一非核心节点 NC_{id} 转发一个分组时，不改变 A-map 信息，但是添加自己的身份识别码如下：($C_{id}+NC_{id}$, A-map, band-id)。假如输入信息中的 ID 等于 3，那么非核心节点 NC_{id} 不添加自己的 ID，并将 vFlag 设为 NULL。

(3) 下行核心节点

一个核心节点接收到一个元信息时，不仅知道源核心节点有哪些分组，而且知道其用来请求重传的传输路径。假如将 vFlag 设为 NULL，那么该核心节点仍然使用 A-map，但是退回到到达早先发送重传请求的有关核心节点的存储路径。

每个核心节点就地维护两个 A-map：表示已成功接收的分组的 myBM，表示已接收和被请求重传的分组的 totBM。

一个核心节点接收到一个输入 A-map (inBM) 时，检查其是否来自一个有效源节点。假如源节点有效，那么接着检查该 inBM 是否携带有某个还未被接收到或者还未被请求重传的分组有效性信息。假如至少有一个有效分组，那么该核心节点生成一个请求 A-map，更新自己的 totBM，然后发送这个请求，启动请求定时器。

对于一个被成功接收的分组，核心节点更新其 myBM 和 totBM。请求定时器结束时，更新 totBM。

核心节点若是在特定时间（核心节点存在定时器，时间值大于 3 个转发跳往返传输时间）内没有接收到其任何上行核心节点发送的 A-map，则直接给默认上行核心节点发送请求，上行核心节点利用其当前 A-map 做出响应。

2. 非核心节点丢失恢复

非核心节点旁听其核心节点的所有发送和重传，一旦观察到其核心节点发送的 A-map 全部比特被置位，则立即进入非核心节点恢复阶段，启动对该核心节点的重传请求。如果非核心节点在核心节点存在定时周期内没有旁听到其发送或者重传，则直接给核心节点发送重传请求，核心节点利用其当前 A-map 做出响应。图 9-6 (c) 表示核心节点和非核心节点之间的丢失检测与恢复。因为核心节点的所有重传均被非核心节点所旁听到，所以排除了相同丢失分组的冗余重传。

9.3.7 其他可靠性语义的支持

前面按照单个分组和多个分组交付以及所有节点 100%交付语义描述了 GARUDA 体系结构。下面针对其他三种可靠性语义描述 GARUDA 体系结构：①对网络某个子区域中的所

有节点进行可靠交付；②对覆盖整个感知区域的最小节点集进行可靠交付；③对网络中 $p\%$ 的节点进行可靠交付。

这三种可靠性语义与所有节点 100%可靠交付语义之间的差异只是网络中要求可靠交付的节点子集不同而已。将要求可靠交付的节点子集的决定问题称为候选资格问题。在所讨论的所有解决方法中，第一个分组总是交付给所有网络节点，随后的分组则根据候选资格来交付。

一般地，使用如下三个公共要素来处理这三种可靠性语义：

① 第一个分组携带应该可靠接收整条消息的合格候选节点信息。例如，对于子区域可靠性语义，第一个分组可能携带一个描述子区域的坐标。

② 只允许已选择自己作为候选节点的那些节点参与 GARUDA 核结构，而 GARUDA 核结构其他方面保持不变（如只有 $3i$ 组带上的节点能够选择自己作为核心节点）。因此在完成 GARUDA 核构建之时，每个独立的候选子图 G_s 部分具有自己的 GARUDA 核。

③ 对于强迫候选资格，使不同部分的 GARUDA 核连接到中心节点。因此， $3i$ 组带上且处于每个部分到达中心节点路径上的非候选节点被强迫作为候选核心节点参与 GARUDA 核构建，以确保连通性。实际上只需要对前述 GARUDA 体系结构进行极少修改就能够实现强迫候选资格。核心组带上的非候选节点除非已经选择自己作为核心节点，否则在转发第一个数据分组时认为自己是非候选核心节点。没有旁听到任何其他候选上行核心节点的下行候选核心节点直接请求上行非候选核心节点成为候选上行核心节点。通过这个过程完成建立一个通过 SPT 连通的各个近似独立 MDS（每个 G_s 部分内）的 GARUDA 核结构。

1. 对子区域内的可靠交付

中心节点很可能只要求将一个查询或者一条消息可靠交付给网络特定子区域内的节点。假定按照坐标描述子区域，不是一般性，又假定子区域是矩形（尽管 GARUDA 对子区域形状没有限制），子区域可以与中心节点所占区域相邻或者不相邻。

中心节点发送的第一个分组携带子区域的坐标描述。网络中的每个节点接收到第一个分组就能够根据自身位置和所需子区域就地决定自己是否为候选节点。一旦确定候选资格后，除非传感器在核心节点组带上，否则传感器的操作就如同全网可靠交付语义（默认操作）一样。但是在默认操作中，传感器只是在发送前旁听到另一个核心节点时才不能选择自己作为核心节点，对此稍作修改为：不管其他条件如何，传感器只要不是候选节点就不会选择自己作为核心节点。这并不意味着这种传感器随后就能够被强迫作为核心节点。

2. 向覆盖感知场的可靠交付

向覆盖感知场可靠交付语义不仅要求可靠交付，而且要求在 WSN 布置时仍然具有固有的冗余度意识。特别是，要求对覆盖整个传感器场的最小传感器子集可靠交付。为了下面的讨论，假定感知范围 S 总是小于或者等于传输范围 R 。

为了实现对覆盖感知场的可靠交付，要求就地确定每个节点的候选资格，要求节点之间相互协商，以便排除覆盖已经被其他传感器所覆盖的区域的传感器的候选资格。在 GARUDA 中，因为核心节点与依赖自己的所有非核心节点直接相邻，在理想情况下非核心节点与最近核心节点（至少是根据其传输范围定义的感知区域的实质“拥有节点”）至少相距 $2R$ ，所以最好配置核心节点来完成这种协调功能。因此在对覆盖感知场的可靠交付中，非核心节点寻

求其核心节点允许其成为候选节点。每个核心节点连续跟踪其覆盖区域（边长等于 $2(S+T)$ 的正方形，核心节点位于中心位置）。只有当非核心节点的覆盖区域不在核心节点的正方形覆盖区域内的时候，核心节点才允许该非核心节点成为一个候选节点。已知 S 和 R ，则核心节点传输范围内不存在其感知覆盖区域位于（即使部分位于）核心节点所定义正方形覆盖区域之外的非核心节点。

所有核心节点间接成为候选节点。在理想情况下，即使核心节点没有与其附近的其他核心节点协商，但是两者相距 $2R$ ，这就意味着核心节点可以选择自己成为候选节点而不会发生与其附近任何核心节点感知区域重叠的问题，因此核心节点间接成为候选节点也是合理的。

3. 对概率子集的可靠交付

对概率子集可靠交付语义要求支持对 $p\%$ 的 WSN 节点的可靠交付。当中心节点打算进行范围感知时，就可能用概率子集可靠交付语义。也就是说，中心节点开始决定只感知场的 25%，然后只是依据初步感知期间检测到的一些触发因素增大感知区域。

在对概率子集的可靠交付中，候选资格的确定也是一个本地过程。一个传感器接收到第一个分组时，以概率 p 选择自己作为候选节点；假如该传感器在核心节点组带上，并且决定不作为候选节点，那么不管其他条件如何，该传感器不会选择自己作为核心节点。

9.3.8 GARUDA的性能

通过 ns-2 仿真评估 GARUDA 在对所有节点 100%可靠交付语义下的性能。对于单个分组可靠交付，比较 GARUDA、ACK 法（使用 ACK 反馈与重传超时进行分组交付）的性能。NACK 不能恢复单个分组丢失。对于多个分组交付，比较 GARUDA、按序交付机制（采用 NACK）、乱序交付机制（采用 NACK）的性能。

1. 仿真环境

对于所有 ns-2 仿真实验：①首先按照栅格方式将 100 个传感器节点布置在 $650\text{ m} \times 650\text{ m}$ 正方形区域内，确保网络连通性，其余节点随机布置在该正方形区域内，中心节点位于一条正方形边中心位置；②每个节点的传输距离 67 m；③信道容量为 1 Mb/s；④每条消息包含 100 个分组，传输速率 25 个分组/秒（单个分组交付除外），组长 1 KB；⑤MAC 协议为 CSMA/CA；⑥ 采用基本泛洪作为路由协议。所有仿真结果是 20 个随机生成拓扑实验结果的平均结果，置信区间 95%。

无线信道误码、传输碰撞均会造成分组丢失。为了评估这两种分组丢失，对于无线信道误码率，选择 5% 的固定分组丢失率；对于传输碰撞，改变网络节点数（因此改变网络密度），从而改变网络竞争程度。

2. 单个分组交付的评估

（1）时延

对于 GARUDA 和 ACK 法，可靠接收单个分组的时延随着传感器的增多而增大，如图 9-7（a）所示。从图 9-7（a）中看到 GARUDA 的交付时延明显小于 ACK 法，其原因在于 WFP 脉冲。

WFP 脉冲实质上是隐含 NACK，因此不会引起网络载荷的提高。由于同样的原因，GARUDA 表现出良好的交付时延与网络节点数之间的比例关系。ACK 法的交付时延高于 GARUDA 的原因在于其直接反馈给发送节点的 ACK，从而导致网络流量的增大，由此引起碰撞的增多。

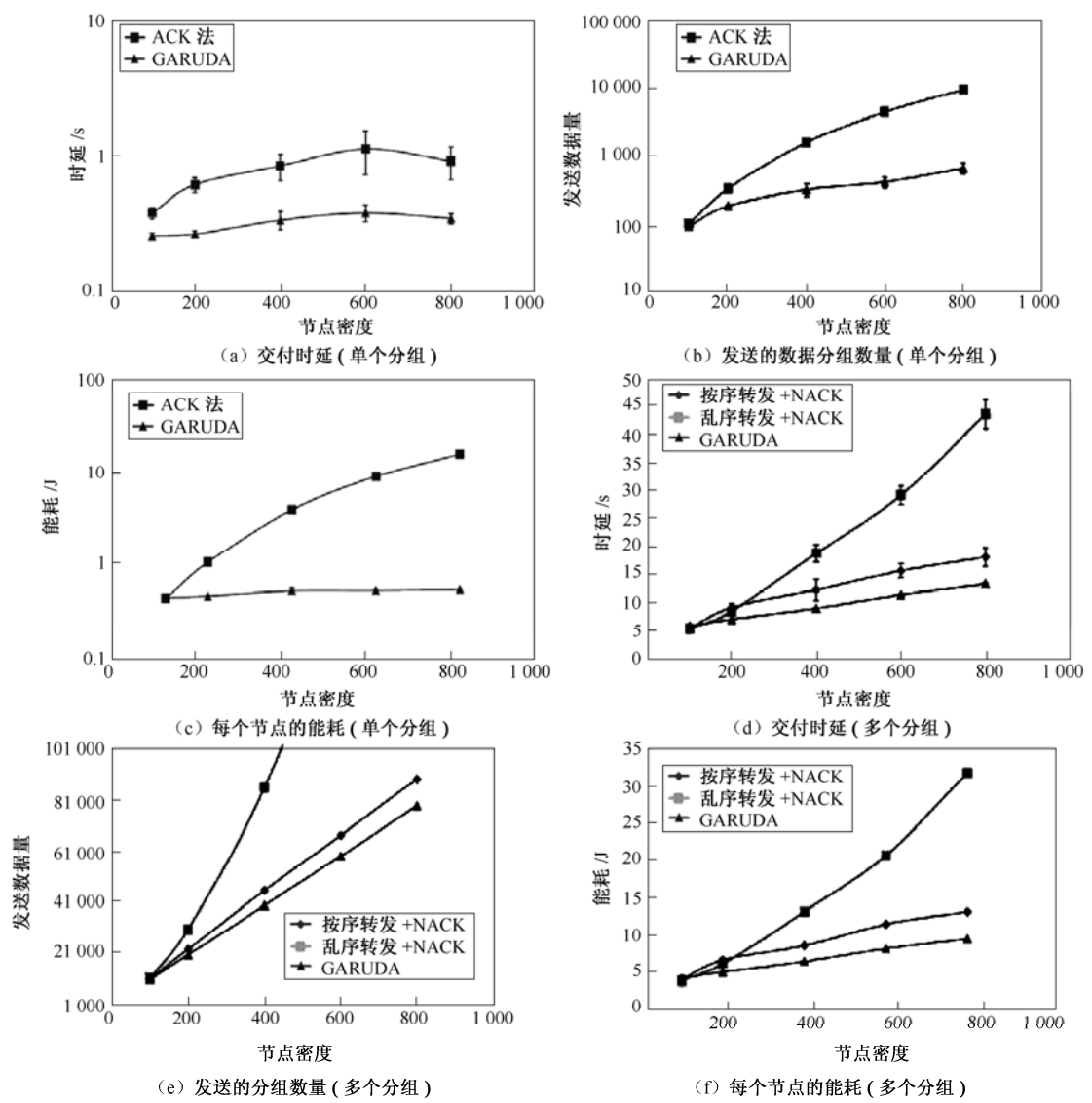


图 9-7 GARUDA 的仿真结果

(2) 发送的数据量

图 9-7 (b) 表示 GARUDA 和 ACK 法发送的数据量。从图 9-7 (b) 中看到 GARUDA 发送的数据量随着网络节点数的增大而线性增大（斜率约等于 1），其原因在于 GARUDA 的隐含 NACK 减轻了有关拥塞引起的分组丢失，泛洪过程固有的冗余度和广播特性确保分组被成功接收，而且即使在发生丢失的情况下也不需要重传。ACK 法发送的数据量稍高于 GARUDA，而且随着网络节点数的增大表现出非线性的增大，其原因仍然在于 ACK 法需要发送 ACK 而导致网络载荷提高，从而导致丢失的增多。

(3) 能耗

每个节点的能耗 (单位为 J), 如图 9-7 (c) 所示。从图 9-7 (c) 中看到 GARUDA 的能耗明显低于 ACK 法, 其原因有两点: 第一, WFP 脉冲持续时间比数据分组发送时间小得多, 可以小到 $15\sim 20\ \mu\text{s}$ 也仍然能够被识别; 第二, WFP 脉冲本身不会遇到任何 WFP 脉冲暴问题, 而是用来解决 ACK 暴问题的。实际上, 能耗随着网络节点数的增大而线性增大。ACK 法存在 ACK 暴问题, 其能耗随着节点密度的增大而增大。

3. 多个分组交付的评估

为了评估 GARUDA 多个分组交付的性能, 实现两个简单的可靠传输协议, 以便分别使用按序转发机制和乱序转发机制, 同时结合基于 NACK 的误码检测和非指定性本地恢复服务器。实验结果如图 9-7 (d) ~图 9-7 (f) 所示。

(1) 时延

100%交付时延与网络节点数之间的变化关系如图 9-7 (d) 所示。从图 9-7 (d) 中看到 GARUDA 的交付时延明显低于按序转发和乱序转发, 并且 GARUDA 交付时延优势随着网络密度的提高而提高 (尤其对于乱序转发), 其原因有两个: 第一, 采用本地特指服务器能够减少发送数据量, 因此比采用非特定服务器有优势; 第二, 采用本地特指服务器不存在 NACK 暴问题, 提高了网络的空间复用, 因此比采用乱序转发有优势。当节点密度较高时, 乱序转发 (含 NACK) 存在 NACK 暴问题, 因此其交付时延及其增大速度明显高于 GARUDA 和按序转发。尽管 GARUDA 核结构采用乱序交付, 但是通过发送的每个分组携带核心节点 A-map, 从而允许其他依赖节点等待核心节点恢复所有丢失分组, 然后才重传请求, 因此排除了 NACK 暴问题。

(2) 发送的数据量

三种可靠传输协议的发送数据量如图 9-7 (e) 所示。GARUDA 的发送数据量最低, 按序转发 (含 NACK) 次之, 乱序转发 (含 NACK) 最高。当节点密度等于 400、600、800 时, GARUDA 的发送数据量比按序转发 (含 NACK) 少 10% 以上, 但是比乱序转发 (含 NACK) 少 80% 以上。其主要原因还是在于 GARUDA 使用特指恢复服务器而没有使用非特指恢复服务器及其 A-map 结构传播机制。

(3) 每个节点的能耗

从图 9-7 (f) 中看到, GARUDA 每个节点的平均能耗明显低于另外两种可靠传输协议。三种可靠传输协议每个节点的平均能耗均直接正比于发送次数, 发送次数等于每个节点发送的请求个数和数据量之和。GARUDA 的发送次数最少, 因此 GARUDA 每个节点的能耗明显低得多。实际上, 实验结果指出: 对于 800 个节点, GARUDA 每个节点的能耗比按序转发低 30% 左右, 比乱序转发低 80% 左右。

参 考 文 献

- [1] B. Hull, K. Jamieson, and H. Balakrishnan. Mitigating congestion in wireless sensor networks. In ACM SenSys, pages 134-147, 2004.

- [2] Lu, C., Blum, B. M., Abdelzaher, T. F., Stankovic, J. A., and He, T.. RAP: A Real-Time Communication Architecture for Large-Scale Wireless Sensor Networks. In Proc. of the IEEE RTAS Symposium (San Jose, CA, September 2002).
- [3] Luo, H., Lu, S., and Bharghavan, V.. A New Model for Packet Scheduling in Multihop Wireless Networks. In Proc. of the ACM MOBICOM Conf. (Boston, MA, August 2000), pp.87-98.
- [4] Mishra, P., and Kanakia, H.. A Hop by Hop Rate-based Congestion Control Scheme. In Proceedings of the ACM SIGCOMM Conf. (Baltimore, MD, August 1992), pp.112-123.
- [5] Ozveren, C., Simcoe, R., and Varghese, G. Reliable and Efficient Hop-by-Hop Flow Control. In Proceedings of the ACM SIGCOMM Conf. (London, UK, August 1994).
- [6] Yi, Y., and Shakkottai, S.. Hop-by-hop Congestion Control over a Wireless Multi-hop Network. In Proc. of the IEEE INFOCOM Conf. (Hong Kong, June 2004).
- [7] S. Floyd, V. Jacobson, C. Liu, S. Macanne, and L. Zhang. A reliable multicast framework for lightweight session and application layer framing. *IEEE/ACM Trans. Netw.*, vol.5, no.2, pp.784–803, Dec.1997.
- [8] TinyOS Homepage [Online]. Available: <http://webs.cs.berkeley.edu/tos/>
- [9] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. in Proc. 5th Annu. ACM/IEEE Int. Conf. Mobile Comput. Netw., Aug.1999, pp.151–162.
- [10] C-Y. Wan, A. T. Campbell, and L. Krishnamurthy. PSFQ: A reliable transport protocol for wireless sensor networks. in Proc. 1st ACM Int. Workshop Wireless Sensor Netw. Appl.(WSNA), Atlanta, GA, Sep.28, 2002, pp.1–11.
- [11] J. Zhao, R. Govindan, and D. Estrin. Computing aggregates for monitoring wireless sensor networks. in Proc. IEEE ICC Workshop Sensor Network Protocols Appl., Anchorage, AK, May 2003, pp.139–148.
- [12] J. Zhao and R. Govindan. Understanding packet delivery performance in dense wireless sensor networks. in Proc. 1st ACM Conf. Embedded Netw. Sensor Syst. (SenSys), Nov.2003, pp.1–13.
- [13] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker. Complex behavior at scale: An experimental study of low-power wireless sensor networks. Comput. Sci. Dept., Univ. California, Los Angeles, CA, Tech. Rep. UCLA/CSD-TR02-0013, 2002.
- [14] C-Y. Wan. A resilient transport system for wireless sensor networks. Ph.D. dissertation, Dept. Elec. Eng., Columbia Univ., New York, 2005. [Online.] Available: <http://comet.columbia.edu/armstrong/wan-2005.pdf>.
- [15] C-Y. Wan, A. T. Campbell, and L. Krishnamurthy .Pump-Slowly, Fetch-Quickly (PSFQ): A Reliable Transport Protocol for Sensor Networks. in Proc. IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL.23, NO.4, APRIL 2005, pp.862–872.
- [16] BYERS, J. W., LUBY, M., MITZENMACHER, M., AND REGE,A.. A digital fountain approach to reliable distribution of bulk data. In Proceedings of the Special Interest Group on Data Communications (ACM SIGCOMM) (Oct. 1998), pp.56-67.

- [17] F.HARARY. Graph Theory. Addison Wesley Publishing Co.,Oct. 1969.
- [18] Seung-Jong Park, Ramanuja Vedantham, Raghupathy Sivakumar and Ian F. Akyildiz. A Scalable Approach for Reliable Downstream Data Delivery in Wireless Sensor Networks. MobiHoc'04, May 24-26,pp.78-89,2004.
- [19] JOHNSON, D. S.. Approximation algorithms for combinatorial problems. In Journal of Computer and System Sciences 9 (1974), pp.256-278.
- [20] LI, D., AND CHERITON, D. R.. OTERS (On-Tree Efficient Recovery using Subcasting): A reliable multicast protocol. In Proceedings of the International Conference on Network Protocols (ICNP) (Oct.1998), pp.237-245.
- [21] LI, D., AND CHERITON, D. R.. Evaluating the utility of FEC with reliable multicast. In Proceedings of the International Conference on Network Protocols (ICNP) (Nov. 1999), pp.97-105.
- [22] SAVVIDES, A., AND SRIVASTAVA, M. B.. A distributed computation platform for wireless embedded sensing. In Proceedings of the International Conference on Computer Design (ICCD) (Sept. 2002), pp.220-225.
- [23] SIVAKUMAR, R., SINHA, P., AND BHARGHAVAN, V.. CEDAR: A Core-Extraction Distributed Ad hoc Routing algorithm. IEEE Journal on Selected Areas in Communications (Special Issue on Ad-hoc Routing) 17, 8 (Aug. 1999), pp.1454-1465.
- [24] TANG, K., AND GERLA, M.. Mac reliable broadcast in ad hoc networks. In Proceedings of the conference on Military Communications (MILCOM) (Aug. 2001), pp.1008-1013.
- [25] VAZIRANI, V. V. Approximation Algorithms. Springer, May 2001.
- [26] WILLIAMS, B., AND CAMP, T.. Comparison of broadcasting techniques for mobile ad hoc networks. In Proceedings of the international symposium on Mobile Ad Hoc Networking and Computing (ACM MOBIHOC) (June 2002), pp.194-205.
- [27] F. Stann and J. Heidemann. RMST: Reliable data transport in sensor networks. In IEEE SNPA, pages 102-112, 2003.

第 10 章 无线传感器网络数据融合技术

WSN 数据融合技术非常丰富, 不仅涉及信号处理技术, 而且还包括相对简单的网内数据累积技术。本书主要介绍网内数据累积技术。

数据累积定义为多个传感器感知数据的累积过程, 排除冗余传输, 给中心节点提供融合信息。数据累积是 WSN 的有效通信节能技术和重要技术之一。在 WSN 中, 通信开销常常比计算开销高几个数量级。由于传感器节点能量有限, 所以所有传感器直接将其感知数据发送给中心节点的能量效率极差。相邻传感器产生的感知数据常常包含冗余数据, 且高度相关。大规模 WSN 产生的感知数据对于中心节点是非常大的。因此通过网内数据累积排除冗余数据, 只转发从原始感知数据中提取出来的精华信息, 减少发送给中心节点的分组数量, 通常能够节省能量和带宽、降低通信开销。因为降低通信能耗能够延长网络寿命, 所以 WSN 支持网内数据累积非常重要。

下面详细介绍几个典型而各有其特点的 WSN 网内数据累积技术, ①在 WSN 中常用的树状结构累积技术; ②独立于应用的累积技术; ③无结构与半结构的累积数。

10.1 树状结构累积

在树状传感器网络中, 将传感器节点组织成一棵树, 树上的中间节点执行数据累积, 将累积数据的精简表示发送给树根节点。这种数据累积适用于涉及网内数据累积的应用。例如, 核电厂内的核辐射监视应用, 最大核辐射强度提供最为有用的该核厂安全信息。下面简单描述数据累积树的构建方法。

E-Span 是一种能量意识生成树算法, 选择能量最高的源节点作为树根, 其他源节点根据其相邻节点的剩余能量和到达树根的距离从中选择一个相邻节点作为其父节点。

在描述生成树协议之前, 首先作如下两个定义:

- ① 树枝能量: 一根给定树枝上所有非树叶节点的最低能量。
- ② 树能量: 一棵给定树上所有树枝的最低树枝能量。

设 B_y 表示一根树枝上的节点集合, 其树叶节点为 y ; 设 I_x 表示一棵树 (树根在节点 x) 上的节点集合。树枝能量和树能量的数学计算公式如下 (其中 e_k 表示第 k 个传感器节点的剩余能量):

$$\text{树枝 } B_y \text{ 的树枝能量} = \min_{i \in B_y, i \neq y} \{e_i\} \tag{10-1}$$

$$\text{树 } I_x \text{ 的树能量} = \min_{j \in I_x, j \neq \text{leafnode}} \{e_j\} \tag{10-2}$$

10.1.1 分布式生成树算法

一棵生成树就是一张图, 以所有网络节点为顶点, 不包含闭环。构建生成树时, 以身份

识别码 ID 最小的节点为树根，所有其他节点通过最短路径路由连接所选择的树根。要求每个节点交换配置消息。配置消息包含本节点的 ID、所选树根的 ID、到达所选树根的距离（转发跳数）。每个节点找到较小 ID 的树根，或者找到最短路径相邻节点的时候就更新其配置消息。一个节点 u 只要检测到其一个相邻节点 v ，并且接收到 v 发送来的配置消息，就选择 v 作为自己的父节点。必要时利用节点 ID 打断连接。

将上述生成树描述转化为 GetSpan 算法，其伪码如图 10-1（a）所示，其中“single-hop broadcast”表示将一个分组发送给所有一跳相邻节点。第 1、2 行将配置消息交换限制在事件区域内，第 3 行启动配置消息交换以及一个用于生成树维护的加法定时器。第 4 行触发一个无限循环。第 5~9 行允许树根按照 T 秒间隔周期性产生配置消息，将开始失去其最短路径相邻节点的节点复位。第 10~15 行节点自我更新，在节点找到较小 ID 树根或者更好的最短路径相邻节点时转发配置消息。

例如，利用上述分布式生成树协议由图 10-2（a）所示的一组源节点的连通图得到图 10-2（b）所示的一棵生成树。由于没有考虑节点剩余能量而导致这棵生成树最低能量为 3 J。剩余能量最少的节点 1 被选做树根节点，并且还连接 3 个其他子节点。当使用这棵树收集各个源节点的数据时，节点 1 的能耗速率相当高，因此到达出现第一个节点失效时的时间最短。因此，对上述分布式生成树协议稍加修改，得到下面介绍的 E-Span 生成树协议。

```
Define:  $r_n$  表示节点  $n$  选择的树根的 ID  

 $d_n$  表示从  $r_n$  到达节点  $n$  的最短路径  

 $g_n(n, r_n, d_n)$  表示节点  $n$  发送的配置消息  

 $s_n = (n, e_n, r_n, e(r_n), p_n, d_n)$  表示节点  $n$  发送的消息  

 $p_n$  表示节点  $n$  选择的父节点的 ID  

 $t_{recv,n}$  表示节点  $n$  接收到其父节点发送的配置消息所需要的时间  

Initialize:  $g_n$  to  $(n, n, 0) \forall n \in N$   

 $p_n$  to  $n \forall n \in N$   

 $t_{recv,n}$  to 0  $\forall n \in N$   

GetSpan(node ID  $N$ , time  $t$ , timeframe  $T$ )  

1 if ( $n$ =event source) //假如节点  $n$  不是事件源节点，则返回  

2 return;  

3 single-hop broadcast  $g_n$ , and start a timer  $P$  that expires every  $T$  sec;  

4 while true  

5 if  $P$  expires and  $(r_n = n$  or  $t > t_{recv,n} + T)$   

6 set  $g_n$  to  $(n, n, 0)$ ;  

7 set  $p_n$  to  $n$ ;  

8 set  $t_{recv,n}$  to  $t$ ;  

9 single-hop broadcast  $g_n$ ;  

10 if receiving a message  $g_i$  from node  $i$   

11 if  $r_i < r_n$  or  $(r_i = r_n$  and  $d_i + 1 < d_n)$  or  $(r_i = r_n$  and  $d_i + 1 = d_n$  and  $i < p_n)$   

12 set  $g_n$  to  $(n, r_i, d_i + 1)$ ;  

13 set set  $p_n$  to  $i$ ;  

14 set  $t_{recv,n}$  to  $t$ ;  

15 single-hop broadcast  $g_n$  and restart timer  $P$ ;
```

(a) 分布式生成树协议算法

```
Define:  $e_n$  表示节点  $n$  的剩余能量  

 $r_n$  表示节点  $n$  选择的树根的 ID  

 $e(r_n)$  表示节点  $n$  选择的树根的最新更新能量  

 $d_n$  表示从  $r_n$  到达节点  $n$  的最短路径  

 $p_n$  表示节点  $n$  选择的父节点的 ID  

 $s_n = (n, e_n, r_n, e(r_n), p_n, d_n)$  表示节点  $n$  发送的消息  

 $e(p_n)$  表示节点  $n$  选择的父节点的最新更新能量  

 $t_{recv,n}$  表示节点  $n$  接收到其父节点发送的配置消息所需要的时间  

childrenList $_n$  表示节点  $n$  的子节点列表  

Initialize: Change  $(n, e_n, n, e_n, 0, 0) \forall n \in N$   

GetSpan(node ID  $N$ , node energy  $e_n$ , time  $t$ , timeframe  $T$ )  

1 if ( $n$ =event source) //假如节点  $n$  不是事件源节点，则返回  

2 return;  

3 single-hop broadcast  $s_n$  and start a timer  $P$  that expires every  $T$  sec;  

4 while true  

5 if  $P$  expires and  $(r_n = n$  or  $t > t_{recv,n} + T)$   

6 Change  $(n, e_n, n, e_n, 0, 0)$ ;  

7 single-hop broadcast  $s_n$ ;  

8 if receiving a message  $s_i$  from node  $i$   

9 if ( $p_i = n$ )  

10 add  $i$  to childrenList $_n$ ;  

11 else  

12 remove  $i$  from childrenList $_n$ ;  

13 if ( $r_i = r_n$ )  

14 if  $(e_i > e_n)$  or  $(d_i + 1 < d_n)$  or  $(d_i + 1 = d_n$  and  $e_i > e(p_n))$  or  $(d_i + 1 = d_n$  and  $e_i = e(p_n)$  and  $i < p_n)$   

15 Change  $(i, e_i, r_i, e(r_i), d_i + 1, 0)$ ;  

16 else if  $(e_i > e(r_n))$  or  $(e_i > e(r_n))$  and  $r_i < r_n$   

17 Change  $(i, e_i, r_i, e(r_i), d_i + 1, 0)$ ;  

18 if  $(e_i > e(r_n))$  or  $(e_i > e(r_n))$  and  $r_i < r_n$   

19 Change  $(n, e_n, n, e_n, 0, 0)$ ;  

20 if a change is applied  

21 single-hop broadcast  $s_n$ ;  

Change(node  $x$ , energy  $e_n$ , node  $y$ , energy  $e_y$ , distance  $d$ , time  $t$ )  

1 set  $s_n$  to  $(n, e_n, y, e_y, d)$ ;  

2 set  $p_n$  to  $y$ ;  

3 set  $e(p_n)$  to  $e_y$ ;  

4 set  $t_{recv,n}$  to  $t$ ;
```

(b) 分布式能量意识生成树协议算法

图 10-1 分布式生成树协议

10.1.2 E-Span树

E-Span 是一张图，以所有网络节点为顶点，不包含闭环。选择能量最高的节点作为树根，所有其他节点通过最短路径路由连接所选择的树根。树根除了收集数据，还要负责与中心节点协调各条路由。任何其他节点选择能量最多的相邻节点作为其父节点（最短路径通过这个父节点并接收其发送来的消息）。对图 10-2（a）使用 E-Span 算法构成的树如图 10-2（c）

所示：选择能量最高的节点 8 作为树根，所有其他节点通过最短路径路由连接到节点 8。节点 6 有两个最短路径相邻节点 2 和 4，但是只连接能量较高的那个节点（即节点 2）。理由是在数据收集时选择有效资源较多的节点作为父节点。

E-Span 算法的伪码如图 10-1 (b) 所示。配置参数涉及三个参数：配置消息发送节点 u 的剩余能量，节点 u 选择的树根的剩余能量，节点 u 选择的父节点的剩余能量。第 1~3 行启动配置消息交换，将消息交换限制在事件区域内。第 4~7 行允许树根按照 T 秒间隔周期性产生配置消息，将与其父节点失去连接的节点复位。第 8~12 行更新接收节点的子节点列表。第 13~17 行在一个节点接收到其父节点的能量更新之时或者检测到一个更好的最短路径相邻节点或者能量更多的树根之时更新配置消息。第 18~19 行比较接收节点和树根。第 20~21 行在发生变化之后广播配置消息。

但是在 E-Span 中，由于不知道所有源节点提供的连通性全集，所以有些节点仍然会通过树根，路由包含较低树枝能量。结果每个源节点较频繁参与树重构，E-Span 树上中断链路恢复消耗源节点较大一部分可用资源。比如在图 10-2 (c) 中，节点 3 连接节点 6，节点 5 连接节点 3，因此树能量等于 7 J，而不等于 3 J，因此节点 1 和节点 7 的能耗速率较低。反之，节点 3 没有连接节点 6，节点 5 没有连接节点 3，那么节点 1 和节点 7 由于 E-Span 树重构消耗额外能量而功能寿命较短。当源节点比较少时，发生这个问题的概率极低，但是当源节点很多时，则需要采用下面介绍的 LPT 来进行累积。

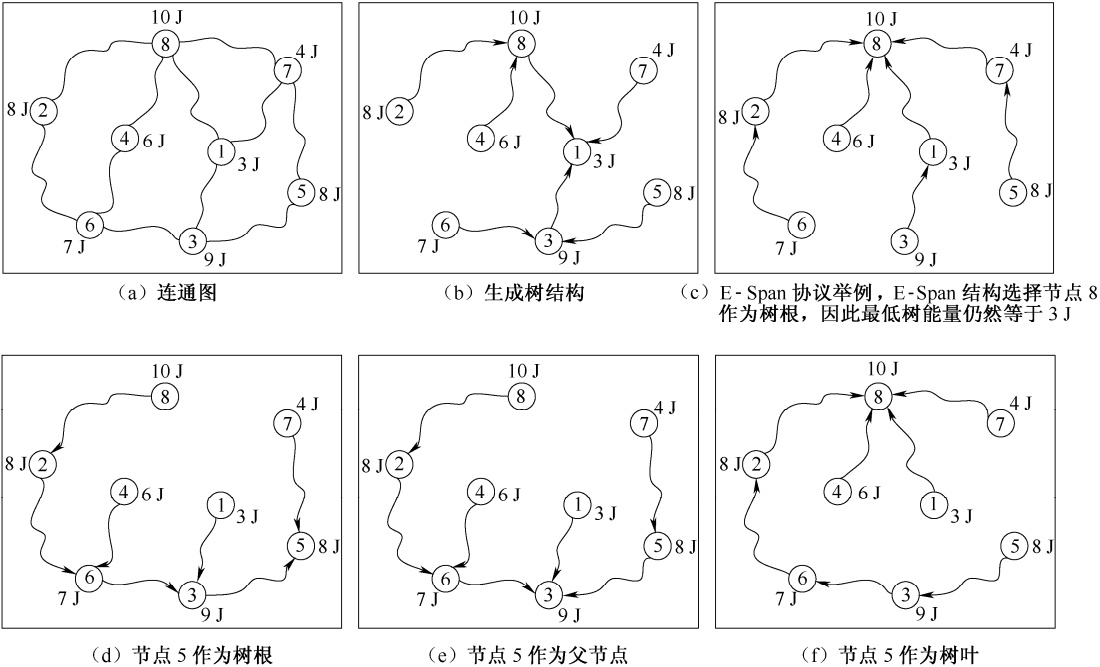


图 10-2 生成树协议与 E-Span 协议的例子

10.2 不受应用约束的自适应数据累积 (AIDA)

美国弗吉尼亚州大学提出的不受应用约束的自适应数据累积 (Application Independent

Data Aggregation, AIDA) 主要是解决无线传感器网络固有的低带宽、能量有限问题, 其目标是实现无线通信信道利用率最大化。AIDA 根据当前本地流量模式动态改变转发节点的数据累积程度, 减轻信道竞争、分组头以及固定长度分组填充比特带来的高开销。

AIDA 不同于依赖应用的数据累积 (Application Dependent Data Aggregation, ADDA) [见图 10-3 (b)]。ADDA 依赖应用层信息, 必须有双向接口, 因此依赖数据中心路由协议。AIDA 可以无缝嵌入到传感器网络通信协议栈中[见图 10-3 (a)], 将 AIDA 累积决策与具体应用隔离开。AIDA 通用性强, 适用于广泛的应用 (数据类型)。此外, AIDA 还可以作为其他数据累积策略的补充策略[见图 10-3 (c)], 节省大量低层协议通信时间。

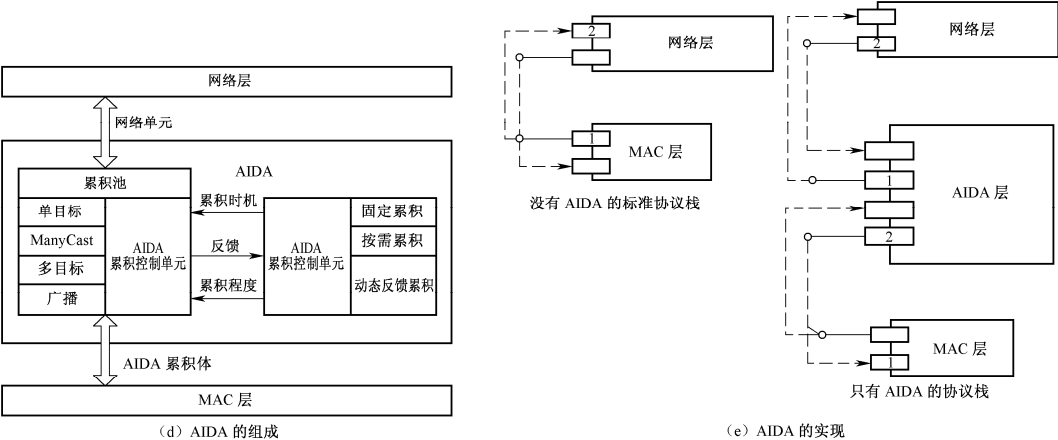
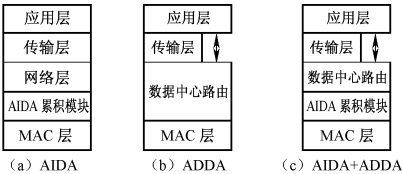


图 10-3 AIDA 的组成与实现

10.2.1 AIDA协议概述

将 AIDA 累积模块安排在数据链路层与网络层之间, 累积依次串联通过网络单元的各个分组。AIDA 累积模块将各个网络单元组合成单一的 AIDA 有效载荷, 以便减少应答和降低信道竞争期间的开销。网络单元中不采用数据语义。根据自适应反馈分组传输时间安排协议做出累积决策, 分组传输时间安排协议按照流量变化状况动态控制累积程度。

10.2.2 AIDA体系结构

AIDA 基本结构如图 10-3 (d) 所示。将 AIDA 功能分成两个部分: 一个功能单元是对网络分组 (单元) 进行累积和去累积, 称为 AIDA 累积功能单元; 另一个功能单元是 AIDA 累积控制单元, 用于自适应控制定时器设置和精确调整所需要的累积程度。AIDA 累积控制单

元是一个基于反馈的自适应组成单元，根据本地当前网络状况在线做出决策，AIDA 工作原理如下：将网络层发送下来的分组送入累积池。根据一次累积串联的分组数量以及这些分组的下一个转发跳接收节点，AIDA 累积功能单元选择一种 AIDA 分组格式（总共四种格式，稍后将介绍）来进行累积，然后将累积结果下传给 MAC 层。AIDA 累积控制单元决定累积的分组数量以及调用累积的时间。

类似于输出流量，MAC 层接收输入流量并将其上传给 AIDA。输入累积体在 AIDA 内被分解成原来的各个网络单元，将恢复出来的每个网络单元上传给网络层更改其传输路由或者上传给应用进行分解和交付。尽管很多累积针对同一个最终目的节点（不在每个中间节点上进行累积和去累积可能效率更高），但是进行去累积是为了确保各层的模块化以及允许网络组成要素独立决定每个网络单元的路由。

将多个网络单元累积成单个 AIDA 累积体进行传输，降低了信道竞争开销（等待/退避）以及控制分组开销（比如 IEEE 802.11 中的 RTS/CTS/ACK，规律性可靠 MAC 中的 ACK），每次累积均要产生这些开销。通过增加单个 AIDA 累积体中的网络单元数[将一个 AIDA 累积体中累积的网络单元数称为累积程度（Degree Of Aggregation，DOA）]，那么每次发送时能够节省 $[(DOA-1) \times \text{竞争时间}] \text{ ms}$ 。

尽管 AIDA 累积功能单元很直观，但却是设计自适应 AIDA 控制单元、在线设置合适定时参数和 DOA 参数的一个复杂研究问题。不同控制方案对系统性能影响很大。

为了保持 AIDA 对其他协议层的透明，采用委托法截获 MAC 层和网络层之间的所有函数调用。委托法假定直接与 MAC 层通信，MAC 层直接与其通信。采用委托法后，AIDA 数据累积层模拟 MAC 层和网络层的接口。根据这种方法得到的 AIDA 协议栈如图 10-3（e）所示。

10.2.3 AIDA控制单元中的累积方案

AIDA 体系结构包括固定累积、按需累积、动态反馈累积，如图 10-4 所示。从基于静态门限的累积决策到最终的动态在线反馈控制机制，全部被综合到 AIDA 体系结构中。基准（不进行累积）用于对比。

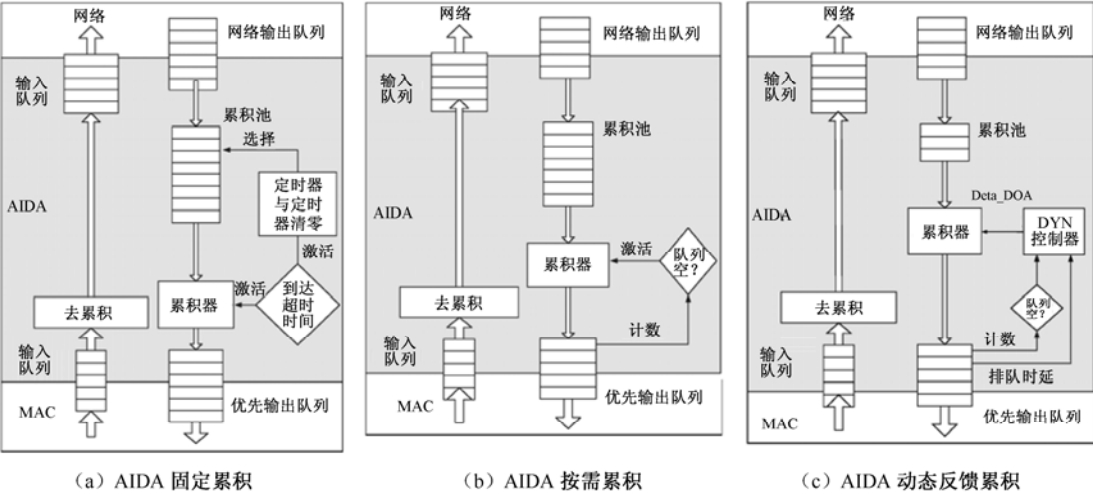


图 10-4 AIDA 累积方案

1. 无累积

若不进行累积（基准），则不作任何修改地使用标准网络协议栈将分组从网络层直接交付给 MAC 层以及将分组从 MAC 层直接交付给网络层。

2. 固定累积

在固定（FIX）累积方案中，每个 AIDA 有效载荷累积的网络单元数是固定的（ $DOA=N_{\text{fixed}}$ ）。将固定数量的网络单元累积完毕后，将 AIDA 有效载荷下传给 MAC 层，以备发送。采用定时器（ T_{fixed} ）技术确保在一定时间范围内执行累积（不管网络单元的数量），从而确保网络单元在发送前等待确定的时间。固定累积的实现框图如图 10-4（a）所示。

3. 按需累积

为了防止不必要的每个转发跳时延，按需累积监视 AIDA 输出队列，确保队列中总是有 AIDA 有效载荷下传给 MAC 层出队列和发送。当 MAC 层可以发送时，AIDA 层不会阻碍网络单元（除非已达到最大 MAC 单元长度），以便达到较高 DOA。只有在合适时间（已建立起输出消息队列或者传输媒介忙而防止 MAC 层访问信道）才会进行 AIDA 层数据累积。按需累积实质上提供透明累积，不会引起消息时延开销。按需累积的工作原理如图 10-4（b）所示。按需累积是反应式累积，被动测量允许 DOA 随着流量模式的变化而变化。当流量极轻时，极少建立起输出消息队列，因此不执行累积。随着流量的增加，输出消息队列的长度增大，因此 DOA 相应增大。

从图 10-4（b）可以看到，按需累积只需要测试输出消息队列是否为空的简单监视逻辑。实现代码简单，是传感器节点所需要的。对比简单的所有分组全部出队列，按需累积通过累积一系列网络单元，每个累积只有一个 MAC 分组头，降低了分组头开销。

4. 动态反馈累积

AIDA 的最终解决方案就是动态反馈累积（DYNamic Feedback Aggregation, DYN）。DYN 综合按需累积和固定累积，动态调整 DOA 门限值（ N_{DYN} ）。如图 10-4（c）所示，DYN 通过监视 AIDA 输出队列来决定其可用性，同时收集排队延迟数据，排队延迟数据影响 AIDA 有效载荷的发送。DYN 累积机制运用这些信息（可用性和收集的数据）以及控制论动态调整累积程度（ $DOA=N_{\text{DYN}}$ ），使 MAC 时延收敛到某个确定点上。DYN 开始时将 N_{DYN} 设为 1。在轻网络流量下，DYN 按照默认的按需累积机制工作，只要有分组，就将其交付给 MAC 发送队列。随着网络流量的增大以及竞争导致发送延迟，DYN 反馈环调整准入门限（ N_{DYN} ），允许按照较高累积程度进行发送。

直观上，可以采用基于试探法（而不是基于理论基础）的算法调整 DOA，影响分组所经历的 MAC 层时延。当 MAC 层时延增大时，DOA 门限值增大，降低至 MAC 层的反馈速率，因此参与信道竞争的节点较少，从而导致 MAC 层时延变小。但是，由于试探反馈控制没有系统动态性信息，因此存在反应过快或者过慢，不能很好适应系统的问题。因此，有必要开发用于 DOA 和 MAC 层之间动态性研究的分析模型，指导反馈控制器开发。

通常运用时隙化方法（如在 ALHOA 和 CSMA 中）分析竞争类协议性能以及建立系统模型。尽管 AIDA 没有假定时隙化 MAC 协议，但是可以采用这种分析方法简化问题的公式表示，建模过程如下。

计算 MAC 时延的通用公式为

$$D_{\text{MAC}}(k) = D_{\text{minimum}} + \# \text{collision}(k) \times D_{\text{reslove}} \quad (10-3)$$

式中， $D_{\text{MAC}}(k)$ 表示分组在时间间隔 $[k, k+1]$ 内经历的 MAC 时延； D_{minimum} 表示无碰撞时的 MAC 时延，是控制环希望达到的性能点； $\# \text{collision}(k)$ 表示一次成功发送在时间间隔 $[k, k+1]$ 内遇到的碰撞次数； D_{reslove} 表示碰撞时延与解决单次碰撞所需时间之和，通常被认为是常数。式 (10-3) 建立起 MAC 层模型。构建一个 AIDA 分组的等待时延随流量而定，在 MAC 建模过程中不应该考虑。

假定在某个时间间隔，来自不同传感器节点的 $N(k)$ 个分组准备待发送。统计上，AIDA 平均向下交付 $N(k)/\text{DOA}(k)$ 个分组，主动竞争信道。 $\text{DOA}(k)$ 等于所有竞争该信道的节点的 DOA 的平均值。一个分组在该时间间隔内被发送的概率表示为 τ ， τ 是 MAC 协议类型的函数。一个输出分组至少与剩余 $N(k)/\text{DOA}(k)-1$ 个分组中的一个分组的发送重叠，则该输出分组遇到一次碰撞。因此，平均碰撞概率 p 计算如下

$$p = 1 - (1 - \tau)^{N(k)/\text{DOA}(k)-1}, \quad N/\text{DOA} \geq 1 \quad (10-4)$$

每次成功传输所需要的平均发送次数为

$$E(\# \text{collisions} + 1) = 1/(1 - p) \quad (10-5)$$

将式 (10-4) 代入式 (10-5)，得到每次成功传输所遇到的碰撞次数期望值为

$$E(\# \text{collisions}) = \frac{1}{(1 - \tau)^{N(k)/\text{DOA}(k)-1}}, \quad N/\text{DOA} \geq 1 \quad (10-6)$$

联合式 (10-3) 和式 (10-6)，可得到 DOA 和 MAC 层时延之间的近似相关性，即

$$D_{\text{MAC}}(k) = [D_{\text{minimum}} - D_{\text{reslove}}] + D_{\text{reslove}}(1 - \tau)^{1 - [N(k)/\text{DOA}(k)]} \quad (10-7)$$

由于 D_{minimum} 、 D_{reslove} 和 τ 与 DOA 无关，所以对式 (10-7) 求差分，得到系统小信号模型为

$$\begin{cases} D_{\text{MAC}}(k+1) = D_{\text{MAC}}(k) + \frac{\lambda_1}{\text{DOA}(k)^2} \lambda_2^{\frac{1}{\text{DOA}(k)}} \Delta \text{DOA}(k) \\ \text{DOA}(k+1) = \text{DOA}(k) + \Delta \text{DOA}(k) \end{cases} \quad (10-8)$$

式中， $\lambda_1 = D_{\text{reslove}}(1 - \tau)N(k)\ln(1 - \tau)$ ， $\lambda_2 = (1 - \tau)^{-N(k)}$ 。

由于 λ_1 和 λ_2 与 DOA 无关，所以可以认为 λ_1 和 λ_2 在小信号控制模型附近是常数。近似模型[见式 (10-8)]不是用来计算各种 DOA 设置下的 MAC 时延，而是用来设计 AIDA 反馈控制器。根据所选 MAC 协议的特性可以推导出 λ_1 和 λ_2 的值，从而可以建立特定的模型。但是，由于 MAC 独立性缘故，所以根据式 (10-8) 设计一个通用的 AIDA 反馈控制器，即

$$\Delta \text{DOA}(k) = G(k) \times e(k) \quad (10-9)$$

式中， $G(k) = P_{\text{DOA}} \times \text{DOA}(k)^2$ ， $e(k) = [D_{\text{MAC}}(k) - D_{\text{minimum}}]$ 。

在式 (10-9) 中， P_{DOA} 是一个执行参数，用于设置 DOA 变化与 MAC 时延控制误差之间的增益。因此，AIDA 实质上被模拟为一阶系统。对于稳定性分析，只要求式 (10-9) 中的增益 $G(k)$ 有界而不必是恒定的。

10.2.4 AIDA累积功能单元

图 10-3 (d) 中的 AIDA 累积功能单元负责网络单元的累积和去累积。根据 AIDA 参数集和模块的当前状态, AIDA 累积功能单元包括四种类型的累积: 单目标、ManyCast (多点到一点)、多目标、广播。

① 假如 AIDA 控制单元准备累积时只有一个网络单元 (比如发生超时), 那么 AIDA 累积功能单元采用单目标方式将该网络单元发送给特定相邻节点。在这种情况下, 不执行累积。

② 假如被累积的所有网络单元是发送给相同的下一个转发跳节点的, 那么 AIDA 累积功能单元采用 ManyCast 方式将累积体发送给该转发跳节点。

③ 当被累积的各个网络单元具有不同的下一个转发跳地址时, AIDA 累积功能单元利用稍复杂的多目标方式发送累积体, 以便利用无线通信的广播特性。在这种情况下, AIDA 不管每个网络单元的接收相邻节点而将各个网络单元合并成一个累积体, 采用 MAC 广播地址作为目的地址。发送节点的每个相邻节点接收到多目标分组后对其进行去累积, 确定自己是否为累积体有效载荷某个部分的接收节点。

④ 当被累积的所有网络单元都是广播消息时, AIDA 累积功能单元采用广播方式发送累积体。

尽管理论上单一分组格式 (多目标) 足够支持上述所有累积, 但是针对每种累积单独制定的分组格式能够减小 AIDA 分组头长度和节省带宽, 这对于资源有限传感器网络非常有益, 从而证明通过 AIDA 分类增加的复杂性很小。

10.2.5 AIDA分组格式

与大多数通信栈分层结构一样, AIDA 按照分组头形式给分组增加信息元。AIDA 分组头定义累积格式, 位于所有被累积网络单元前面, 被封装在 AIDA 数据单元中, 用于去累积、分接以及对适当网络层协议的无缝交付。AIDA 数据单元被下传给 MAC 层发送。然后根据一个节点的交付情况就可以利用 AIDA 分组头来验证所使用的特定累积机制 (在这种情况下提供多种累积选择)、评估去累积后的 AIDA 有效载荷结构, 以及有可能分解、分接以及将每个网络单元交付给适当的网络层模块。

通过累积网络有效载荷, AIDA 减少了 MAC 层发送的分组数量, 因此实际上降低了总分组头开销。AIDA 分组头通用格式如图 10-5 (a) 所示。

1. 所有类型的标志

AIDA 分组头的第一个组成域就是长度 8 bit 的标志 (Flag) 域, 用于说明所累积的所有网络单元的有关信息。标志域包含一个 2 bit 的类型子域、一个 2 bit 的协议子域、一个 4 bit 的接收节点/UNIT 数量子域, 具体说明如下:

① 类型子域: 说明本 AIDA 分组的类型 (单目标分组、ManyCast 分组、多目标分组、广播分组)。

② 协议子域: 说明 AIDA 应该将各个网络单元分接至相应的网络层。

③ 接收节点/UNIT 数量子域: 说明随后的分组头数量。对于单目标、ManyCast、广播

流量，将本子域设为本累积体包含的网络单元数量。对于多目标流量，将本子域设为接收本累积体部分内容的相邻节点数量。

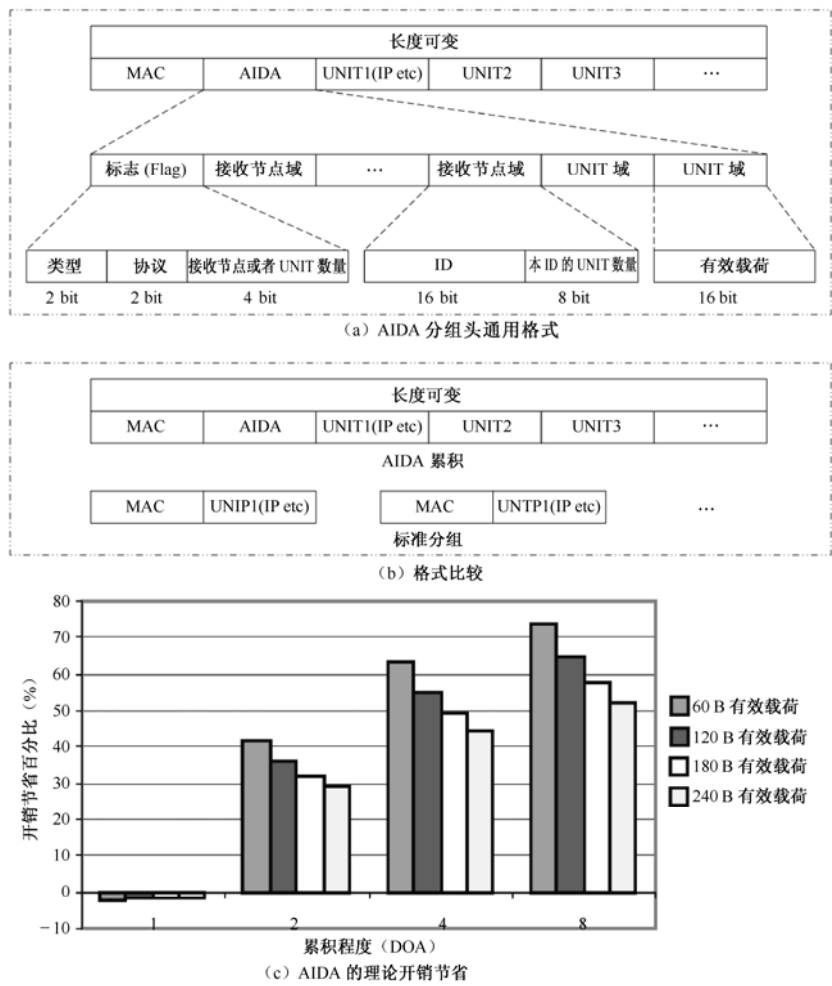


图 10-5 AIDA 分组格式、AIDA 理论开销节省

2. 多目标类型接收节点域
- 只有多目标 AIDA 分组才使用接收节点域。每个接收节点域包含一个 ID 子域，用于说明预定接收节点，随后是本累积体所包含的网络单元数量，这些网络单元就是发送给本 ID 指定的相邻节点。对于单目标、ManyCast、广播 AIDA 有效载荷，不需要区分接收节点，所以不需要使用接收节点域。
- ① ID 子域: 长度 2 B, 用于指定接收特定数量网络单元的节点的本地唯一的身份识别码。
 - ② 本 ID 的 UNIT 数量子域: 长度 1 B, 用于说明累积网络单元数量，这些累积网络单元的接收节点就是本 ID 子域指定的相邻节点。
3. 单元域

在去累积期间运用单元域 (UNIT) 来划分网络单元之间的边界。单元域由一个长 2 B 的

• 276 •

有效载荷子域组成，用于说明每个网络单元的长度。对于单目标流量，不需要识别边界，所以不使用单元域。

10.2.6 AIDA分组头开销分析

尽管 AIDA 引入了新的分组头，但是 AIDA 通过将若干个网络单元累积成一个 MAC 有效载荷，实际上降低了总分组头开销。例如，IEEE 802.11 MAC 分组头长 28 B。若不采用 AIDA，那么发送 N 个网络单元，则总分组头开销为 $28 \times N$ B。若采用 AIDA，则将总分组头开销降低至 $28 + \text{AIDAHeaderSize}$ B，AIDAHeaderSize 表示 AIDA 分组头长度。只要 N (DOA) 大于 1，那么 AIDA 就能有效降低发送期间引入的总分组头开销。

根据前述 AIDA 分组格式评估网络单元累积期间引入的分组头开销。为了比较，图 10-5 (b) 给出了采用和不采用 AIDA 的分组结构。

① 单目标只使用标志域，因此产生 1 B 开销。

② ManyCast 分组和广播分组除了 1 B 标志开销之外，还必须区分多个网络单元的边界，因此平均每个网络单元 $(2+1/N)$ B 开销， N 表示 AIDA 有效载荷中累积的网络单元数量。

③ 对于多目标分组，需要识别多个下一个转发跳节点地址，因此平均每个网络单元 $(2+1/N+3/M)$ B 开销， N 含义同②， M 表示每个下一个转发跳节点的网络单元平均数量。

10.2.7 AIDA节省分析

直观上，对任何发送添加分组头信息都会增加单个分组的传输时间。因此，下面分析将多个上层有效载荷累积成单个累积体进行一次发送时每次发送开销的节省情况。通过分析 AIDA 分组头结构，就能够看到发送单目标、ManyCast、广播分组时的差异。为了更好地理解累积带来的好处以及在不同流量模式下比较各种累积程度，通过理论分析来评估关于发送时间的开销。假设对特定 DOA 进行最佳累积而不会引入等待网络层有效载荷的额外开销。另外还评估不考虑碰撞和退避时的开销节省情况，碰撞和退避是导致最终使用 AIDA 的两个因素。

不存在信道竞争时，对于任意 MAC 层，单个发送节点和单个接收节点之间通信的分组传输开销等于该 MAC 层获取和建立每次发送时所消耗的时间与本次消息发送时间之和乘以发送次数。为了维护 MAC 层的独立性，设 MAC 层发送准备时间为 M ms。对于 IEEE 802.11 MAC 协议，传输开销包括信道侦听、RTS、CTS、ACK 以及控制分组间的非连续等待时间。对于以 R B/s 速率发送长度为 S 的网络单元，AIDA 分组头开销为 H 字节数，DOA 等于所累积的分组个数。利用下式计算开销 C_{AIDA} (单位：ms)

$$C_{\text{AIDA}} = M + (S \times \text{DOA} + H) \times R \quad (10-10)$$

不采取累积发送 DOA 个分组的开销 C_{None} 为

$$C_{\text{None}} = (M + S \times R) \times \text{DOA} \quad (10-11)$$

因此，开销节省百分比 η_{COST} 计算如下：

$$\eta_{\text{COST}} = \left(\frac{C_{\text{None}} - C_{\text{AIDA}}}{C_{\text{None}}} \right) = \left[1 - \frac{S \times R}{M + S \times R} \right] - \frac{M + H \times R}{M + S \times R} \times \frac{1}{\text{DOA}} \quad (10-12)$$

从式 (10-12) 中看到：当不能忽略 MAC 层开销 M 时， η_{COST} 随着 DOA 增大而提高。图 10-5 (c) 给出了 AIDA 在 IEEE 802.11 MAC 协议、200 kb/s 无线传输速率的理论开销节省

百分比。AIDA 有效载荷下传给简化 IEEE 802.11 MAC 协议，该 MAC 协议进行空闲信道侦听、RTS/CTS 握手，接着发送每个数据分组，数据分组要求应答确认。控制分组长 11 B。竞争还包括 5 ms 空闲侦听时间，根据 MICA 技术规范选定的 DIFS、SIFS 分别为 10 ms、5 ms。

图 10-5 (c) 说明了理论节省时间。比较发送单个 AIDA 累积体 (包含 DOA 个网络单元、一个 MAC 分组头) 的时间以及发送 DOA 个独立分组 (没有 AIDA 分组头信息也不进行累积) 的时间，就能够计算出理论节省时间。从图 10-5 (c) 中可以看到：随着 DOA 的增大，时间节省百分比急速增大；随着 AIDA 有效载荷的增大，相对节省时间减少。当数据发送时间占总发送时间主要部分时，就会出现上述情况。从图 10-5 (c) 中还可以看到：当 DOA=1，AIDA 不进行任何累积时，开销等于单个字节数据，实际上没有增加传输时间。

10.2.8 AIDA的性能

利用 GloMoSim 仿真 AIDA。仿真参数配置如表 10-1 所示，这些参数大都是根据伯克利 MICA Mote 传感器技术规范来选择的。分析的性能指标有端到端时延、能耗、MAC 控制分组数量、DOA、AIDA 控制开销。在三类流量模式、总共 72 种不同流量载荷下仿真研究这些性能指标，观察 AIDA 在各种流量下的自适应能力。曲线上的每个数据点是 10 次实验 (不同种子) 的平均结果，从而确保 95% 的置信区间。

表 10-1 仿真参数配置

路 由	GF
MAC 层	简化 IEEE 802.11 DCF
物理层	RADIO-ACCNOISE
传播模型	TWO-RAY
带宽	40~200 kb/s (默认值 200 kb/s, 其他专门指定)
有效载荷	32 B
地形	200 m×200 m
Mote 传感器数量	100
节点布置	均匀
无线传输距离	40 m

根据不同累积方案以及没有累积支持的标准协议栈评估 AIDA，认识和了解没有使用应用信息的有关数据累积。比较无累积、FIX、按需累积、DYN 的性能。实验表明 DYN 反馈是各种参试流量下的最佳解决方案。

1. 流量设置

根据请求和检索语义建立传感器网络的典型通信模式 (点到点、多点到一点、多点到多点)，用于传感器节点和查询实体之间的数据交付。当一个活动节点检测到某个需要报告给远端实体的活动时，就会出现点到点通信。一个查询实体要求整个传感器场周期性报告时，就会采用多点到一点通信模式。比较常见的是，同时运行多个应用，相互交替传输各个流量，这就是多点到多点交叉流量模式。仿真重点就是这三种典型通信模式，如图 10-6 所示。

在测试点到点通信模式时，在仿真地形左下角布置一个节点，该节点给仿真地形右上角的一个节点发送单个 CBR 流，两个节点相距 6~7 个转发跳。在测试多点到一点通信模式时，仿真地形左边 10 个节点均给仿真地形右边中间的一个查询节点发送 CBR 流。在测试多点到多点通信模式时，仿真地形左边 5 个节点均分别给仿真地形右上角查询节点和右下角查询节点发送 CBR 流。每个 CBR 流的发送速率按递增方式逐渐增大，测试 AIDA 在不同流量载荷下的性能。

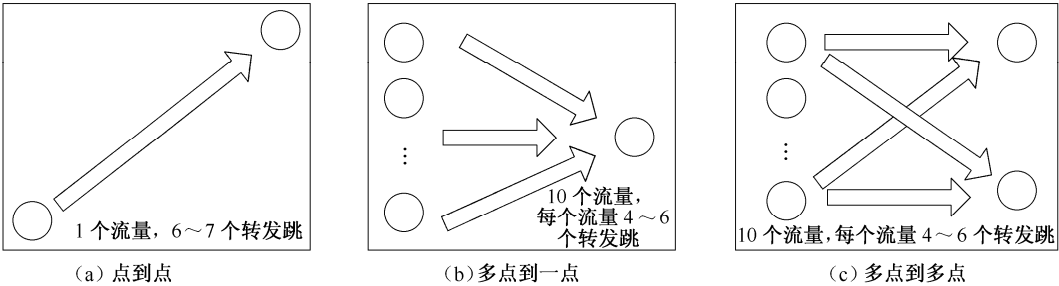


图 10-6 仿真流量载荷设置（每种模式 24 个流量，随机布置 10 个）

2. AIDA端到端时延

(1) 不同累积方案的端到端时延

AIDA 的主要目标是既节能又不影响端到端时延。AIDA 不仅不会引起端到端时延的增大，而且在深度累积下反而会减少 MAC 层使用的控制分组。

图 10-7 (a) ~图 10-7 (c) 给出了三种流量模式的端到端时延与流量载荷的实验变化曲线，从中可以看到：若不采取累积措施，则 CBR 的端到端时延随着总流量的逐渐增大而明显增大，这是多跳无线网络的典型情况（因为多跳无线网络的信道竞争程度明显高于单跳无线局域网）；在轻流量载荷[比如低于 3 个分组/秒，见图 10-7 (b)]下，除了固定累积，其他所有累积方案的端到端时延很小（70~100 ms）。固定累积引起端到端时延增大的原因是：即使信道可用，但是为了获取其特定累积程度，固定累积仍然保存分组；发送速率越低，固定累积需要的等待时间就越长。而按需累积和 DYN 尽可能将分组发送出去，因而不会额外增大端到端时延。按需累积由于其反应式自适应机制而端到端时延表现良好。DYN 根据 MAC 层输出分组时延动态调整所需 DOA，因而端到端时延表现最好。在重流量载荷下，DYN 通过降低发送速率减少信道竞争节点，因而有利于端到端时延。从图 10-7 (c) 中可以看到：当流量载荷特别重（每个流量高达 8.5 个分组/秒）时，DYN 能够将端到端时延降低 80%（相对于不采取累积时的端到端时延）。

(2) 各种可用带宽设置下的端到端时延

根据每种带宽设置分别选择流量，从未饱和流量到饱和和流量比较 AIDA 各种累积方案的端到端时延。

实验结果说明：不论有效带宽设置如何，DYN 的端到端时延总是优于其他累积方案。其主要原因是：DYN 根据当前流量反馈信息能够比其他累积方案更加有效地累积分组和安排分组的传输时间。基于这些实验结果及其分析而得到结论：DYN 相对于其他累积方案的端到端时延改善与有效带宽设置互不相关，但是绝对性能改进可能变化。

(3) 固定累积在不同 DOA 设置下的端到端时延

图 10-7 (d) 给出了固定累积方案在不同 DOA 设置下的端到端时延, 揭示了固定累积方案的缺点, 说明了为什么需要动态自适应。从图 10-7 (d) 中可以看到: 对于每种流量模式, 固定累积在单个 DOA 值下的时延性能表现差。

一方面, DOA 值大, 则固定累积在轻流量载荷条件下时延表现差。例如, 若 DOA 大于 1, 当流量载荷为每个流 0.5 个分组/秒或者更低时, 会引入额外时延。DOA 值越大, 则拥塞减轻, 但是分组等待发送的时延增大。另一方面, DOA 值小, 则固定累积在重流量载荷条件下时延表现差。例如, 如图 10-7 (d) 所示, 若 DOA=1, 当流量载荷为每个流 10 个分组/秒或者更高时, 固定累积的端到端时延约是 DOA=2 时的端到端时延的 2 倍。

固定累积对流量信息不敏感。为了在轻流量和重流量下同时进行最优化, 按需累积和 DYN 提供在线自适应, 能够分别根据这些流量模式被动和主动地改变 DOA 值, 从而表现出更好的总体时延性能。

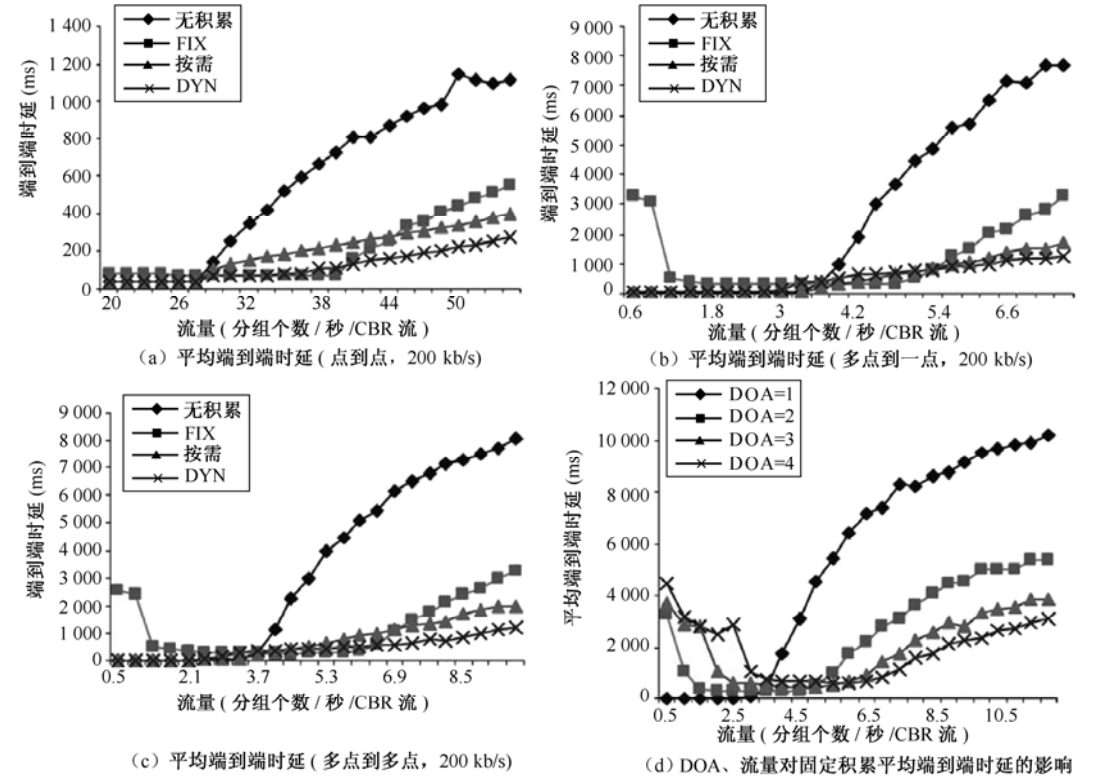


图 10-7 AIDA 端到端时延仿真结果

3. AIDA的节能

采用发送能耗评估 AIDA 性能。发送能耗正比于所发送的比特数量, 能够充分概括和反映 AIDA 其他性能指标, 比如总分组头开销、碰撞次数、总发送比特数等。

由于传感器节点能量资源非常有限, 所以传感器节点以最低能耗进行无线通信对延长传感器网络寿命至关重要。AIDA 采用以下几种方法节能: ①AIDA 将信道竞争分散、跨越多个

网络单元,从而减轻 MAC 信道竞争开销;②AIDA 使用的 MAC 控制分组较少,抑制了拥塞,减少了碰撞次数,从而重传次数较少;③传感器网络协议通常采用固定长度分组(比如 TinyOS 网络协议),因此存在不必要的填充比特开销。在 AIDA 仿真中为了支持可变长度分组,充分利用前两种方法,在三种流量模式的 24 种逐渐递增流量有效载荷下,测试每个交付分组的平均发送能耗。

图 10-8 给出了发送能耗实验结果。实验结果说明:若不采用 AIDA,则发送能耗最高。例如,如图 10-8 (c) 所示,当每个流量速率约等于 6 个分组/秒时,发送能耗并没有达到 DYN 发送能耗的 2 倍。固定累积总是累积 2 个分组后才发送,所以在轻、重流量载荷下的发送能耗几乎是恒定的。但是在固定累积中,设置 DOA 值,没有考虑拥塞程度,因此在重流量载荷下的发送能耗高于按需累积和 DYN。例如,如图 10-8 (c) 所示,当每个流量速率约等于 8 个分组/秒时, DYN 发送能耗约等于固定累积发送能耗的 20%。

图 10-8 (d) 表示固定累积在不同 DOA 值下每个交付分组的发送能耗。图 10-8 (d) 说明:对于固定累积,采用较大 DOA 值, AIDA 能够实现较高节能。但是,在轻流量载荷下,较大 DOA 值会额外增大端到端时延。因此,考虑到端到端时延,增大 DOA 值不是总是有利的。

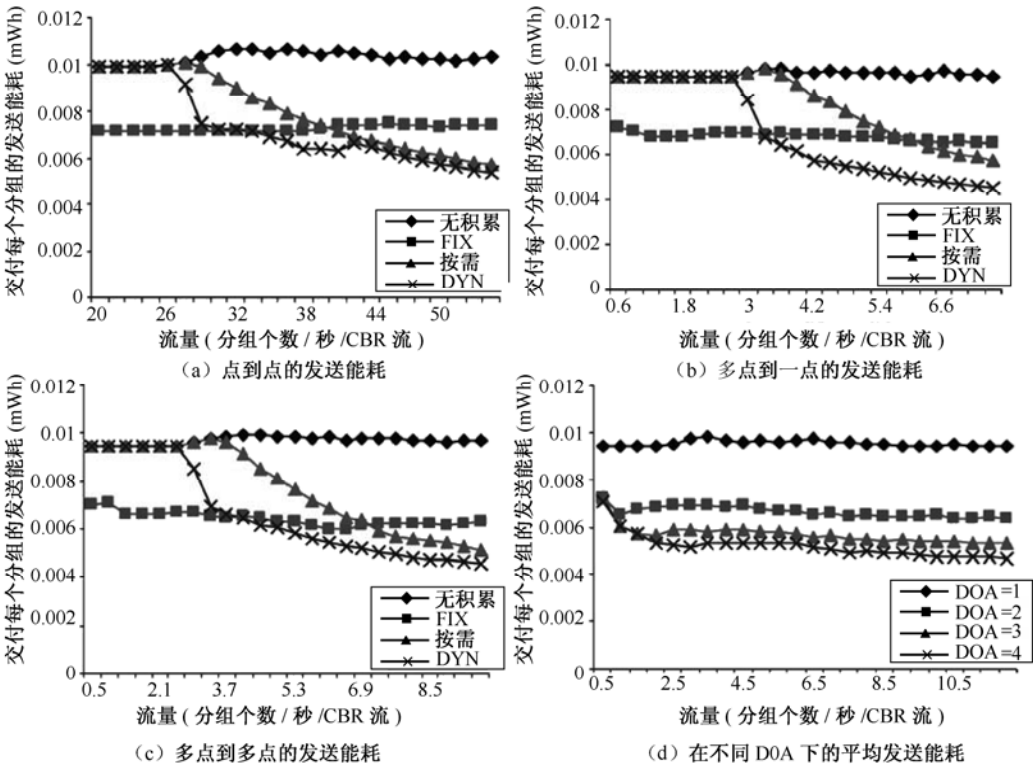


图 10-8 各种流量模式下的发送能耗

10.3 无结构累积法与半结构累积法

前面概述了分群网络的累积方法,还较详细地介绍了树状网络中的累积方法,这些累积方法都是基于某种网络结构(分群结构、树状)的累积,因此将其称为结构累积法。在数据

采集应用中，比如环境和栖息地监视应用，节点周期性向中心节点报告其感知数据，流量模式保持不变，因此适合采用结构累积法，并且结构维护开销低。但是在基于事件的应用中，比如入侵检测和生物危险品检测，无法事先知道源节点，因此采用固定结构的累积方法不能有效累积数据，采用结构动态变化的累积方法却存在结构维护开销高的问题。

为各种累积功能建立最佳数据累积结构是一个难以解决（NP-hard）的问题。对于动态事件，当感知一个事件的一个节点组随着时间而变化时，高效数据累积网络结构的计算与维护开销很高。结构法对等待时间敏感，节点根据等待时间等待其子节点的分组，然后累积分组，将累积结果转发给中心节点。此外，尽管可以采用试探法建立数据累积结构，但是在累积结构中，传输流量表现出会聚特性（即在分群结构中，节点将其分组发送给群首，群首是流量的汇聚点；而在树状结构中，节点将其分组发送给父节点，父节点是流量的汇聚点），而跟会聚传输流量模式有关的另一个问题是基于结构的数据累积协议性能差。会聚传输流量会引起激烈竞争，导致最短路径树（Shortest Path Tree, SPT）的分组丢失率较高，因此丢失分组较多、时延增大，结果难以规定分组的固定发送顺序，从而对结构方法的数据累积性能产生不利影响。通常在树结构中，必须按照某个固定顺序将分组从树叶发送至树根，实现最大程度累积。丢失的分组不仅使最佳累积结构变成次佳累积结构，而且浪费分组发送能量（丢失分组不能传递到达中心节点）。

那么是否可以不需要建立和维护某种网络结构进行数据累积，解决上述问题呢？答案是肯定的。美国俄亥俄州州立大学计算机科学与工程系为基于事件的传感器网络提出了一种无结构数据累积法。这种累积方法没有预先构建的累积结构，也不需要明确维护某种结构，以数据意识任意组播（Data Aware Anycast, DAA）法满足数据累积的空间收敛条件，以随机等待（Randomized Waiting, RW）法满足数据累积的时间收敛条件。但是这种无结构法的性能没有网络规模可扩展性能力。

针对无结构法缺乏网络规模可扩展性的问题，俄亥俄州州立大学计算机科学与工程系又提出了一种叫做半结构法的累积方法。半结构法的目标是在源节点附近实现数据累积，但是不用明确建立移动事件结构。在源节点附近累积分组对于减少发送次数非常关键。不采用明确结构进行累积降低了累积结构建立与维护的开销。

半结构法综合利用了结构法和无结构法的优点，适用于规模极大的传感器网络。半结构法以无结构法（如 DAA 法和 RW 法）为基础，由 DAA 和动态转发两个阶段组成。在第一个阶段，运用 DAA 将分组转发给一个选定的节点（将这种节点称为累积器），然后累积器累积各个分组。在 DAA 中是将分组发送给中心节点，而在半结构法中是将分组转发给累积器。在第二个阶段，在定向非循环图树（Tree on Directed acyclic graph, ToD）结构上转发剩余还未被累积或者被部分累积的分组，以便进一步累积这些分组。下面首先描述无结构法（如 DAA 协议和 RW 协议），然后描述 ToD 上的动态转发协议。

10.3.1 数据意识任意组播（DAA）

DAA 是无结构分组累积协议，能够提高空间收敛和时间收敛。发送期间的空间收敛和时间收敛是累积的两个必要条件。分组必须在相同时间发送给同一个节点才能够被累积。结构法将分组发送给累积树的父节点，父节点等待其所有子节点的分组，然后才发送累积后得到的分组，从而实现空间收敛和时间收敛两个条件。无结构法没有直接交换消息，因此节点不

知道将分组发往何处，也不知道等待多长时间才进行累积。因此，提高空间收敛和时间收敛对于提高累积机会非常关键。

1. 空间累积

空间收敛的实现方法是采用任意组播将分组转发给能够实现累积的节点。任意组播是一种路由协议，根据某些路由参数将分组转发给一个最佳节点、一个任意节点或者一组目标节点。因为位于发送覆盖范围内的节点能够接收到发送分组，所以节点知道自己是否能够累积发送分组并将该信息通知其他节点，任意组播机制允许发送节点将分组转发给其中任何一个节点。将分组发送给能够完成累积的节点，减少了网络中的剩余分组，因此减少了总发送次数。

DAA 法以 MAC 层任意组播为基础，决定接收发送的下一个转发跳节点。任意组播要求首先通过 RTS-CTS 控制分组交互，然后才发送分组。定义累积 ID（Aggregation ID，AID）关联两个能够被累积的分组。RTS 包含发送分组的 AID，以及能够做出 CTS 响应且具有相同 AID 和分组的相邻节点。这里采用时戳作为 AID。因此，同时产生的两个分组就很可能被累积。因为可能存在多个接收节点能够累积一个分组，所以接收节点随机延迟其 CTS 发送，以避免发生 CTS 碰撞。发送节点的两个相邻节点之间的干扰能够预防多个 CTS 发送。节点若是在其随机时延期间接收到任何分组，则取消其 CTS 发送。两倍以上于传输范围的干扰范围足以产生所需要的干扰。图 10-9 给出了 IEEE 802.11 单目标和任意组播中随机延迟 CTS 响应之间的区别：在 IEEE 802.11 中，接收节点接收到 RTS 后立即回送 CTS；而在 DAA 中，接收节点接收到 RTS 后延迟一段随机时间后再回送 CTS，接收节点 2 的 CTS 随机推迟时延较大，因而在接收到接收节点 1 回送的 CTS 后取消其 CTS 发送。

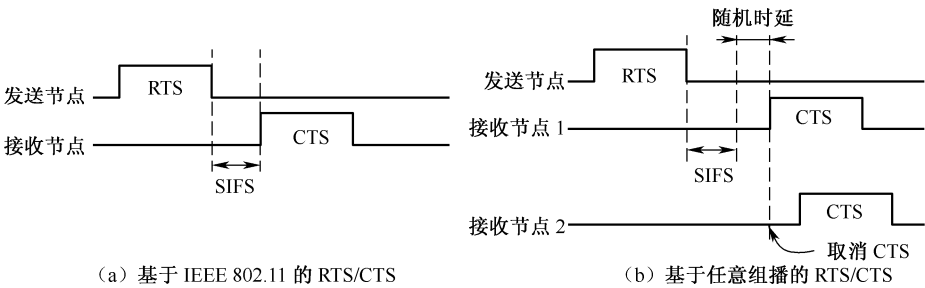


图 10-9 单目标与任意组播的差异

采用 DAA 法，分组能够收敛到少数几个节点上，将这些节点称为累积点。动态选择累积点，不需要直接构建累积树。但是，分组有可能被转发到远离中心节点的节点上。为了降低出现这种情况的概率，设置优先级，给离中心节点较近的节点分配高于发送节点的 CTS 发送优先级。CTS 发送优先级越高，则发送 CTS 前的随机时延越短。可以采用任何邻近参数，包括地理距离和转发跳数。

为了进一步提高累积，仍然使用 DAA 将分组从累积点转发至中心节点，但是需要对 DAA 做如下强化：假如节点没有分组需要累积，但是离中心节点较近，那么以低于有分组需要累积的节点的发送优先级回送 CTS。因此，分组遇到机会时仍然能够被累积，否则分组被贪婪地朝中心节点转发。尽管大多数发送分组经过 DAA 累积后能够朝中心节点方向传递，但是

从实验中观察到有些发送分组由于累积的缘故而朝远离中心节点的方向传递。下面将详细讨论 CTS 优先级以及距离参数。

各个节点分得不同的 RTS 响应优先级。三类优先级如下：

- ① A 类：接收节点具有与 RTS 指定的 ID 相同的分组，并且离中心节点的距离小于发送节点离中心节点的距离。
- ② B 类：接收节点具有与 RTS 指定的 ID 相同的分组，但是离中心节点的距离大于发送节点离中心节点的距离。
- ③ C 类：接收节点没有 RTS 指定的 ID 的分组，但是离中心节点的距离小于发送节点离中心节点的距离。

假如接收节点没有 RTS 指定的 ID 的分组并且离中心节点的距离大于发送节点离中心节点的距离，那么接收节点不回送 CTS。对应于能够响应 RTS 的这三类相邻节点，为其 CTS 分组预留三个时隙，CTS 发送优先级按照 A 类、B 类、C 类依次下降，如图 10-10 (a) 所示。同类节点选择小时隙发送 CTS，以避免与同类其他节点发生碰撞。在同类节点之中，离中心节点较近的节点具有较高 CTS 发送优先级。CTS 实际发送时间可能大于小时隙或者时隙。根据其优先级，采用时隙和小时隙将 CTS 发送的启动错开。根据相邻节点间的干扰假设条件，只有第一个 CTS 发送才会成功，因为由于干扰的缘故其他 CTS 发送将被取消。

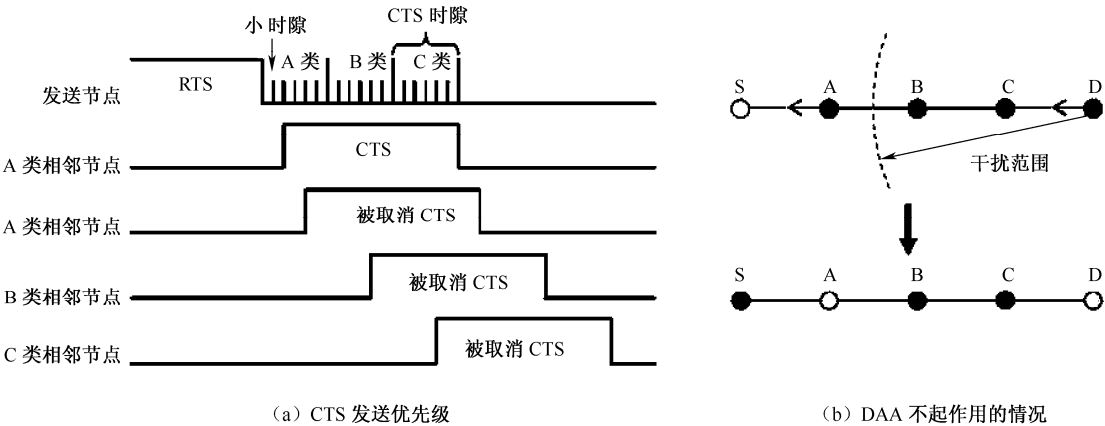


图 10-10 CTS 发送优先级、DAA 不起作用的情况

在 DAA 中，为了设置 CTS 发送优先级，节点必须知道自己是否比发送节点离中心节点更近。CTS 发送优先级用于选择 CTS 时隙和位于 CTS 时隙内的小时隙。可以采用地理距离来比较两个节点离中心节点的远近。节点必须知道自己、相邻节点以及中心节点的位置。RTS 可以包含相邻节点的位置信息，因此接收节点通过 RTS 能够获知发送节点的位置。在网络布置期间，相邻节点之间可以相互交换位置信息。DAA 也可以与其他参数（如转发跳数）一起使用。采用转发跳数时，利用节点与中心节点之间的转发跳数来衡量节点的远近。

每个转发跳均使用 DAA，从而在至中心节点的传输路由上尽可能早地实现分组累积。因为不存在计算累积结构，所以事件移动性不会影响 DAA 的性能。DAA 动态选择传输链路和下一个转发跳节点，因此能够容忍干扰和节点失效问题，从而在不可靠网络中具有极高的强

壮性。但是，DAA 可能不能累积空间隔离（大于一个转发跳）、由 MAC 层一个紧接着一个转发的分组，此时需要采用时间收敛技术进一步提高累积性能。

2. 时间累积

累积的第二个条件（时间收敛）要求分组在相同时间出现在相同节点上，即使对于在相同路由上传递的分组，无结构累积也不能保证对分组进行累积。假如分组发送次序导致分组不能从时间上在中间节点相遇，那么可能会限制累积的优点。分组发送次序受若干因素控制，包括来自其他流和相同流的干扰。

在移动事件触发的网络中，节点无法事先知道事件触发了哪些节点而有分组发送，因此，节点无法知道是否应该等待其上行节点以及对累积分组应该等待多长时间。假定退避时间比分组发送时间小得多。对于结构法，相距几个转发跳的分组，即使处在相同路由上，也仍然可能被逐个转发，直至到达中心节点。为了进一步说明这个问题，考虑一个简单网络拓扑：所有节点呈一字形排列，如图 10-10 (b) 所示。假定无线信号能够干扰两个转发跳内的节点。假如节点 D 首先发送，其间节点 B 和 C 保持静默。因此，不存在与 A 竞争信道的节点。尽管 C 发送 CTS，信道对于 A 不是空闲的，但是 A 只退避一段短时延（小于分组发送时间），之后侦听信道为空闲。由于不存在竞争，所以 A 发送其分组，A 发送的分组不会与来自其上行节点的其他分组进行累积。当分组相距一跳远以上时或者当分组沿着相同路由传递时，DAA 对于提高累积不再起作用。

根据到达中心节点的距离而采用确定性等待时间，使离中心节点较近的节点在发送前等待较长时间能够避免上述问题和提高分组累积机会；但是，所有分组（无论何处产生、无论离中心节点远近）存在固定时延，并且等待时间引起的时延正比于网络规模。在大规模网络中，这种时延很大，因而不能接受。当事件接近中心节点时，接近中心节点的节点选择长时延是不必要的。在大规模网络中采用随机时延避免长时延，同时提高累积的机会。

RW 是实现时间收敛的一种简单技术，因此在源节点采用 RW 技术，对源节点产生的每个分组人为地引入时延，提高时间收敛。每个节点产生一个新分组需要发送时，首先从 $0 \sim \tau$ 中随机选择一个时延， τ 表示最大时延，然后按照这个时延推迟分组发送。在图 10-10 (b) 中，假如节点 A 的时延大于节点 D、节点 B 和 C 的时延小于 A 或者 D，并且 A 和 D 之间的时延差大于分组从 D 到 A 的传输时间，那么 D 的分组能够在 A 被累积。注意：采用 RW 技术后，若数据采样时间小于 τ ，那么分组可能被乱序发送。

τ 的最佳值依赖事件覆盖区域大小，即分组产生节点之间一个转发跳距离的最大值。最大转发跳数增大，最大时延 τ 也应该增大，两个节点选定的时延之差随着增大。假如两个时延之差太小，那么即使下行节点时延较大，但是由于传输时间大于时延差，所以分组不会被累积。最大时延 τ 太大，分组端到端传输时间也太大。若应用不能容忍时延，则必须使用小值 τ ，这样就会失去 RW 技术的优点。

但是，DAA 不能保证所有分组被累积成一个分组。没有经过累积就从源节点发送至中心节点的分组越多，浪费的能量越多。当网络规模非常大、源节点离中心节点很远时，这个效果变得更加严重。因此，当 DAA 不能再累积分组时，不能将分组直接转发给中心节点，而是采用如下描述的 ToD 上动态转发技术进一步累积分组。

10.3.2 ToD上的动态转发

在第二阶段采用预构造结构法实现进一步累积。假如希望将所有分组累积成一个分组，那么必然需要利用某种结构，引导所有分组传递给单个节点。采用直接消息交换动态构建累积结构开销高。因此，使用内含计算的预构造结构，并使该结构保持相对较长时间不变（几个小时或者几天）。但是，采用固定结构存在长伸展问题。以图 10-11（a）为例：预先计算树结构，其中灰色节点是源节点。假如事件 A 触发源节点产生分组，那么由于这些分组在树上可以立即被累积，所以这个固定的树结构能够正常完成累积。但是，假如事件 B 触发源节点产生分组，那么即使这些节点互为相邻节点，这些分组也仍然不会被累积。采用 ToD 上动态转发机制可避免这个长伸展问题。

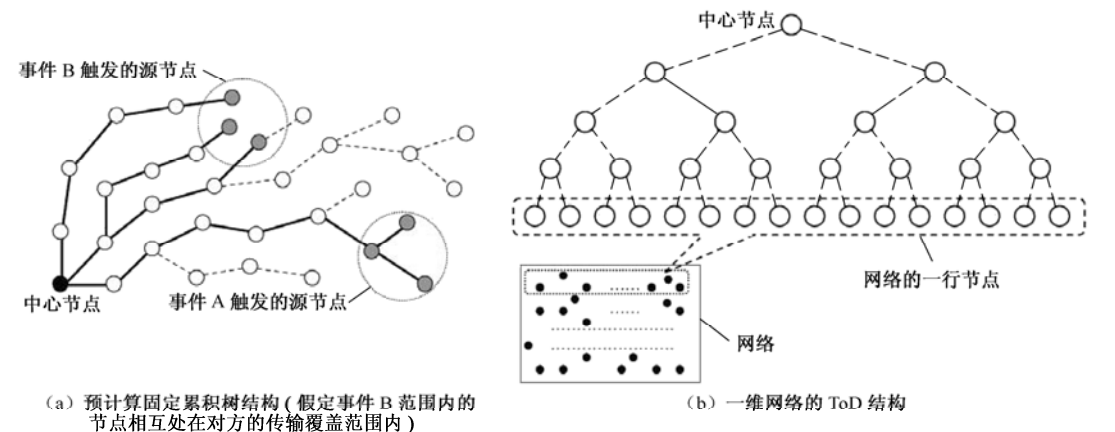


图 10-11 累积树结构与一维网络 ToD 结构

1. 一维网络中的ToD

为了说明 ToD 概念，首先描述一维网络（单行节点）的 ToD 结构，如图 10-11（b）所示。假定同一行节点能够与其相邻节点进行单跳通信。

定义一个蜂窝为一个边长为 Δ 的正方形， Δ 大于一个事件覆盖区域的最大直径。将网络分成一个一个的蜂窝；再对蜂窝进行分群，每个分群由若干个蜂窝组成，将这种分群称为 F-分群（即一级分群）。F-分群必须足够大，能够覆盖一个事件所能覆盖的蜂窝。当只考虑一维网络蜂窝时，一个 F-分群由两个蜂窝组成。每个 F-分群有一个群首，群首被称为 F-累积器（一级累积器）。F-分群内节点与其 F-累积器可能相距几个转发跳远。每个 F-分群内所有节点将其分组发送给自己的 F-累积器，然后每个 F-累积器建立一条到达中心节点的最短路径。因此，累积结构是最短路径树，树根为中心节点，树叶为 F-累积器，将这种累积树称为 F-树（一级树）。

除了 F-分群，还要对蜂窝进行第二种分群，即 S-分群。S-分群也必须足够大，能够覆盖一个事件所能覆盖的蜂窝，而且必须与 F-分群交替，这样才能够覆盖处于不同 F-分群中的相邻蜂窝。每个 S-分群也有一个群首，群首被称为 S-累积器（二级累积器）。每个 S-累积器建

立一条到达中心节点的最短路径，从而在网络中建立第二棵最短路径树（将其称为 S-树），树根为中心节点，树叶为 S-累积器。被一个事件触发的附近所有蜂窝要么位于同一个 F-分群内，要么位于同一个 S-分群内。动态转发利用这个性质避免长伸展问题。

完成 S-树构建后，自行连接相互重叠的 F-分群、S-分群的 F-累积器和 S-累积器。比如在图 10-12 (c) 中，F-分群 4 与 S-分群 3、S-分群 4 重叠，所以 F-累积器 F4 连接 S-累积器 S3 和 S4。因此，联合 F-树和 S-树后创建一张定向非循环图，即 ToD。

图 10-12 (a) 表示 F-树结构。树叶节点就是蜂窝，一对相邻蜂窝构成一个 F-分群，每个 F-分群有一个 F-累积器，各个 F-累积器构成 F-树。图 10-12 (b) 表示 S-树结构。不在同一个 F-分群中的一对相邻蜂窝构成一个 S-分群，每个 S-分群有一个 S-累积器，各个 S-累积器构成 S-树。图 10-12 (c) 表示 ToD 结构。每个 F-累积器连接两个与其 F-分群重叠的 S-分群的 S-累积器，这个结构叫做 ToD。ToD 中的 F-累积器采用动态转发协议将分组转发给中心节点，或者根据分组的源节点而通过 S-树上的一个 S-累积器。

节点首先采用 DAA 方法累积尽可能多的分组。当 DAA 不能再进一步累积分组的时候，节点将其分组转发给其所在 F-分群的 F-累积器。假如一个事件只触发了一个 F-分群内的节点，那么就由该 F-分群的 F-累积器累积该事件的分组，然后利用 F-树将累积结果转发给中心节点。假如一个事件覆盖几个 F-分群，那么将该事件的分组转发给所覆盖的每个 F-分群的 F-累积器。因为假定一个事件的覆盖范围不会大于一个蜂窝的范围，所以位于 F-分群边界上的事件只会触发位于该 F-分群边界上的蜂窝中的节点。通过构建 S-分群，位于 F-分群边界上的相邻蜂窝属于同一个 S-分群。因此，F-累积器可以利用从所收分组中收集的信息选择最适合进一步累积的 S-累积器，可以从分组中包含的传输流源节点获取这些信息。通常分组包含这些信息；否则，单独使用 4 bit 表示分组的源节点。

考虑图 10-12 (c) 中的例子。因为一个事件最多能够覆盖两个蜂窝，所以一个事件覆盖的两个蜂窝要么位于同一个 F-分群中，要么位于同一个 S-分群中。假如一个事件位于同一个 F-分群中，那么该 F-分群的 F-累积器累积该事件的分组。例如，假如一个事件覆盖蜂窝 A 和 B，那么 F-累积器 F1 知道没有其他 F-累积器有分组需要累积，并且采用 F-树转发累积后的分组。假如事件覆盖的两个蜂窝位于两个不同的 F-分群，那么相应的两个 F-累积器只接收其中一个蜂窝的分组，然后根据发送分组的那些蜂窝推断哪个 F-分群可能有分组。例如，假如一个事件覆盖蜂窝 C 和 D，那么 F-累积器 F4 只接收 C 的分组。因此，F4 要么知道只在 C 中发生该事件，要么知道该事件覆盖 C 和 D。因为 S-分群 4 覆盖 C 且与 F-分群 4 重叠，所以 F4 可以将分组转发给 S-累积器 S4。假如 F-累积器 F5 只接收 D 的分组，那么 F5 也将其分组转发给 S4。因此，在 S4 上累积该事件的分组。

需要注意的是：这里并没有明确指定位于网络边界上的蜂窝属于哪个 S-分群。这种蜂窝若是不与任何其他 F-分群相邻，则不必属于任何 S-分群，或者也可以与其相邻蜂窝同属于一个 S-分群。

一维网络 ToD 具有如下性质：

性质 1：对于一维网络 ToD 中的任意两个相邻节点，或者在一级累积器（F-累积器）累积其分组，或者在二级累积器（S-累积器）累积其分组。

一个事件触发节点产生分组只存在三种情况。假如事件只触发一个蜂窝内的节点产生分组，那么该蜂窝内的节点全部属于同一个 F-分群，这个 F-分群的 F-累积器累积其全部成员节点产生的所有分组。

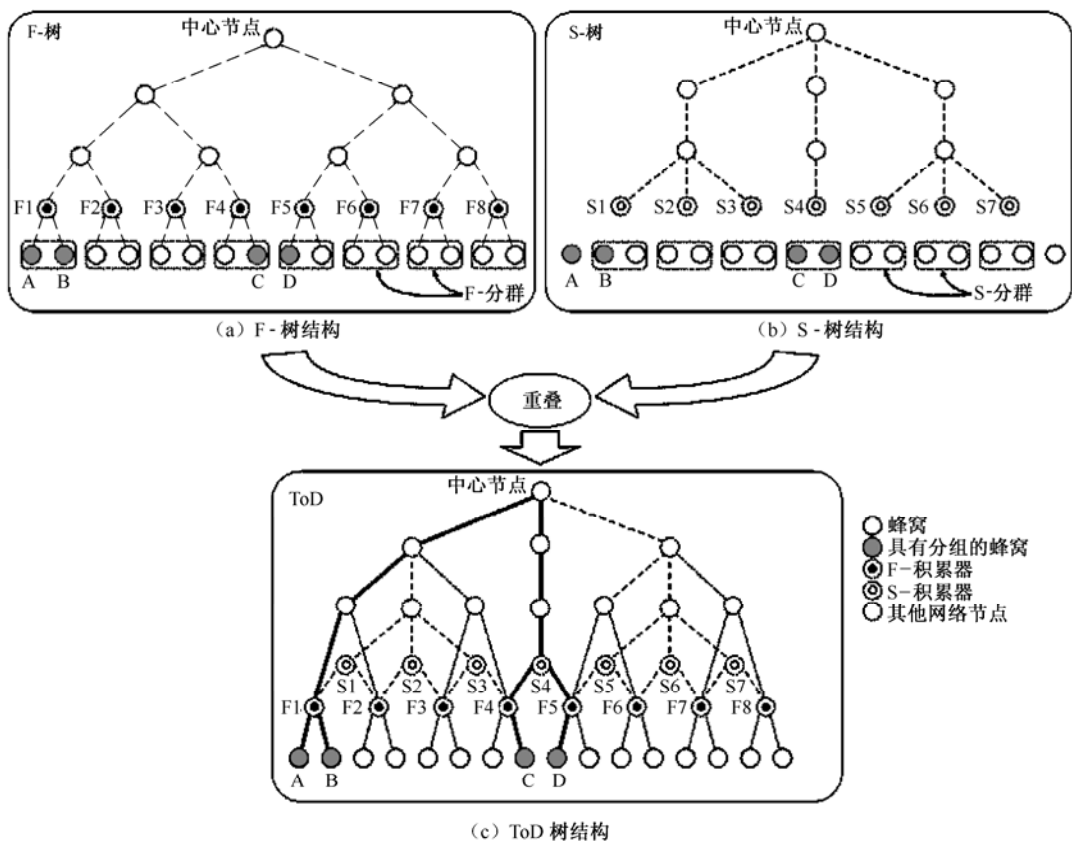


图 10-12 F-树、S-树、ToD 的结构

假如一个事件触发两个蜂窝内的节点产生分组，并且这两个蜂窝属于同一个 F-分群，则同样由这个 F-分群的 F-积累器负责累积这些节点产生的分组。

假如一个事件触发两个蜂窝内的节点产生分组，并且这两个蜂窝分别属于两个不同的 F-分群，但是因为 S-分群和 F-分群是交替的，所以这两个蜂窝必定属于同一个 S-分群。一个 F-分群只接收离另一个 F-分群（并且该 F-分群也有分组）较近的蜂窝产生的分组。因此，F-积累器可以将分组转发给 S-积累器，由后者完成相应的累积。

因为一个蜂窝覆盖的区域不会小于一个事件的最大覆盖区域，所以一个事件不可能触发两个以上的蜂窝。

2. 二维网络中的ToD

前面通过一行节点的结构描述说明了动态转发的基本思想：每个蜂窝只与一个 F-分群相邻，或者当蜂窝位于网络边界上时没有任何相邻的 F-分群。因此，假如一个事件覆盖两个蜂窝，那么这两个蜂窝要么属于同一个 F-分群，要么属于同一个 S-分群，F-积累器能够推测是将分组转发给 S-积累器，还是直接转发给中心节点。当考虑相邻行中其他蜂窝和 F-分群时，位于一个 F-分群边界上的一个蜂窝可能有多个相邻的 F-分群。假如一个事件覆盖多个蜂窝，且假如 F-分群中的蜂窝与多个 F-分群相邻，那么每个 F-积累器有多个 S-积累器选择。假如

这些 F-累积器选择不同的 S-累积器, 那么其分组就不会被累积。但是可以将一维网络 ToD 上的动态转发思想扩充到二维网络上。所不同的是, 一维网络 ToD 是保证分组在两步内被 F-累积器或者 S-累积器所累积, 而二维网络 ToD 则是保证分组在三步内被累积。

首先定义二维网络的蜂窝和分群如下。为了易于理解, 采用栅格分群进行结构说明。按照前面的定义, 一个蜂窝的大小不会小于一个事件的最大覆盖区域, 一个 F-分群必须覆盖一个事件所能覆盖的所有蜂窝, 对于一维网络 ToD 是两个蜂窝, 对于二维网络 ToD (栅格分群) 是四个蜂窝。因此, 整个网络划分成许多 F-分群, 每个 F-分群包含四个蜂窝。每个 F-分群和 S-分群也分别有一个群首负责完成分组的累积功能。图 10-13 (a) ~图 10-13 (c) 表示一个 5×5 网络的 F-分群结构和 S-分群结构: 图 10-13 (a) 表示网络划分成 5×5 个 F-分群; 图 10-13 (b) 表示每个 F-分群包含四个蜂窝 (比如 F-分群 A 包含蜂窝 A1、A2、A3、A4); 图 10-13 (c) 表示 S-分群必须覆盖属于不同 F-分群同时相邻的所有蜂窝, 每个 S-分群包含的四个蜂窝分别属于四个不同的 F-分群。

因为一个蜂窝 (正方形蜂窝的一条边长) 必须大于或者等于一个事件的最大覆盖区域 (直径), 所以一个事件可能覆盖一个、二个、三个或者四个蜂窝, 如图 10-13 (d) 所示。假如一个事件只覆盖同一个 F-分群区域, 那么该 F-分群的 F-累积器累积分组。因此, 下面只考虑一个事件覆盖属于多个 F-分群的蜂窝。

图 10-13 (e) ~图 10-13 (h) 表示一个 F-累积器在收集其 F-分群内产生的所有分组时可能遇到的四种基本情况。其他情况都是这四种基本情况的不同组合。假如分组是同一个 F-分群内的三个或者四个蜂窝产生的, 那么这个 F-分群的 F-累积器知道其他 F-分群中的节点没有分组, 因此将分组直接转发给中心节点。假如只有一个或者两个蜂窝产生分组, 那么其他 F-分群也可能有分组。假定一个事件的覆盖蜂窝是邻近区域。那么在一个 F-分群内, 图 10-13 (e) 和图 10-13 (g) 不可能同时发生, 图 10-13 (f) 和图 10-13 (h) 也不可能同时发生; 但是在真实环境中, 当第三个或者第四个蜂窝产生的分组被丢失时, 则有可能发生这些情况, 此时其他 F-分群没有相同事件的分组, 因此 F-累积器正好将分组直接转发给中心节点。在图 10-13 (e) ~图 10-13 (h) 中, 每种情形表示 3×3 个 F-分群, 由处于中心位置的 F-分群的 F-累积器作出决策, 灰黑色正方形表示产生分组的蜂窝, 淡灰色正方形表示与灰黑色蜂窝对应的 S-分群。

F-累积器收集到其群内所有分组后, 知道这些分组是哪些蜂窝产生的以及将这些分组转发给最合适的 S-累积器进一步累积。例如, 假如分组只是由一个蜂窝产生的, 如图 10-13 (e) 所示, 那么 F-累积器可以将分组转发给覆盖该蜂窝的 S-分群的 S-累积器。但是, 假如分组是由同一个 F-分群中的两个蜂窝产生的, 那么这两个蜂窝必定位于两个不同的 S-分群中。例如, 如图 10-13 (i) 所示, F-分群 X 的 F-累积器接收两个蜂窝的分组, 就是图 10-13 (e) 和图 10-13 (f) 的组合。F-分群 Y 的 F-累积器也可能接收两个蜂窝的分组, 就是图 10-13 (g)、图 10-13 (h) 或者同时为这两者。F-分群 X 的 F-累积器不知道 F-分群 Y 的 F-累积器遇到的是哪种情况, 因而不知道将分组转发给哪个 S-累积器。为了保证累积, F-分群 X 的 F-累积器将分组转发给覆盖蜂窝 C1 和 C2 的两个 S-累积器, 因此分组至少能够在一个 S-累积器相遇。假如两个 F-累积器接收到其分群内两个蜂窝的分组, 那么为了保证分组至少能够在一个 S-累积器相遇, 这两个 F-累积器必须明确选择这个 S-累积器。选择策略是选择离中心节点较近的 S-累积器。假如分组在第一个 S-累积器相遇, 则不必将分组转发给第二个 S-累积器。假如所收分组来自同一个 F-分群中的两个蜂窝, 则只将分组转发给第二个 S-累积器。

为了保证分组至少能够在一个 S-累积器相遇, 第二个 S-累积器的等待时间必须大于第一

个 S-累积器的等待时间。因此，一个 S-累积器若是只接收一个蜂窝的分组，则因为这个 S-累积器可能是另一个 F-累积器的第二个 S-累积器而需要较长时间等待另一个 S-累积器可能转发的分组。图 10-13 (j) 给出一个例子：一个 F-累积器将分组发送给第一个 S-累积器，然后第一个 S-累积器将分组转发给第二个 S-累积器，同时另一个 F-累积器将分组直接发送给第二个 S-累积器。只要第二个 S-累积器的等待时间充分大于第一个 S-累积器的等待时间，那么分组就能够在第二个 S-累积器被累积。

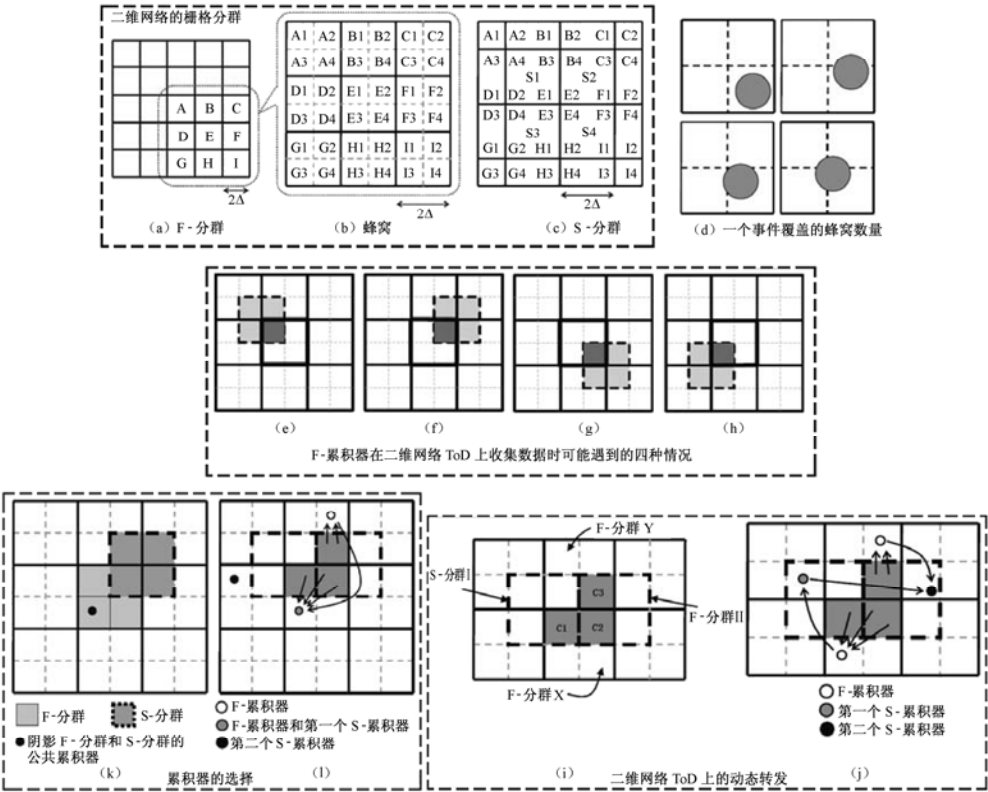


图 10-13 二维网络中的 ToD

性质 2：对于二维网络 ToD 上任意两个相邻节点，其分组或者在 F-累积器，或者在第一个 S-累积器，或者在第二个 S-累积器被累积。

首先定义 F-分群 X 的 F-累积器为 X，S-分群的 S-累积器为 I。

对于一个 F-分群内产生的分组，因为这些分组全部发送给这个 F-分群的 F-累积器，所以由这个 F-累积器完成对这些分组的累积。

假如一个事件触发不同 F-分群中的节点产生分组，那么存在三种情况。第一种情况是，每个 F-分群中只有一个蜂窝产生分组。由于属于不同 F-分群而且又是相邻的蜂窝属于同一个 S-分群，所以对于第一种情况，所有产生分组的蜂窝属于同一个 S-分群，因此其分组也由这个 S-分群的 S-累积器完成累积。

第二种情况是，事件覆盖三个蜂窝 C1、C2、C3，其中两个蜂窝属于同一个 F-分群，另一个蜂窝属于另一个 F-分群。不失一般性，假定 C1、C2 属于同一个 F-分群 X，C3 属于另一个 F-分群 Y。C3 必须与 C1 或者 C2 相邻，这里假定与 C2 相邻。从 ToD 结构知道：C2、C3

属于同一个 S-分群 II，C1 属于另一个 S-分群 I。图 10-13 (i) 给出一个相应例子。因为 C1 和 C2 属于同一个 F-分群，所以首先 F-累积器 X 累积 C1 和 C2 产生的分组，然后将累积结果（分组）转发给 S-累积器 I，再由 S-累积器 I 转发给 S-累积器 II，或者即将累积结果（分组）转发给 S-累积器 II，再由 S-累积器 II 转发给 S-累积器 I，这是因为 C1 属于 S-分群 I，C2 属于 S-分群 II。F-累积器 Y 累积 C3 产生的分组，然后将分组转发给 S-累积器 II（因为 C3 属于 S-分群 II）。F-累积器 Y 的分组只来自 C3 产生的分组，在 S-累积器 II 中等待较长时间，以等待通过另一个 S-累积器转发来的分组；其间，若是 F-累积器 X 首先将分组转发给 S-累积器 II，那么这些分组由 S-累积器 II 完成累积；若是 F-累积器 X 首先将分组转发给 S-累积器 I，那么 S-累积器 I 较快将分组转发给 S-累积器 II，这是因为这些分组来自同一个 F-分群中的两个蜂窝，因此也在 S-累积器 II 上被累积。

第三种情况是，事件覆盖四个蜂窝，其中两个蜂窝属于同一个 F-分群，另两个蜂窝属于另一个 F-分群。不失一般性，假定 C1、C2 属于 F-分群 X，C3、C4 属于 F-分群 Y，C1、C3 相邻，C2、C4 相邻。从 ToD 结构知道：C1、C3 属于同一个 S-分群 I，C2、C4 属于 S-分群 II。F-累积器 X 和 Y 均从 S-累积器 I 和 II 中选择一个离中心节点较近的 S-累积器作为第一个 S-累积器，因此会选择同一个 S-累积器作为第一个 S-累积器，其分组将在第一个 S-累积器被累积。

在上述 ToD 描述中，尽管假定事件的覆盖范围小于一个蜂窝的大小，但是即使无法事先知道事件覆盖范围大小，ToD 动态转发方法仍然能够正常工作，性能表现优于 DAA。这是因为节点只是在第二阶段（此时 DAA 不能再进行累积）才在 ToD 上运用动态转发。因此，在最坏情形下，不过是 ToD 动态转发方法退回到 DAA 法。稍后介绍的测试床实验表明：即使事件的覆盖范围大于一个蜂窝的大小，ToD 仍然能够将 DAA 的性能提高约 27%。

3. 分群与累积节点选择

采用栅格分群方法构建蜂窝和分群。理论上只要满足如下条件，就可以采用任何分群方法（比如六边形分群、三角形分群等）：①蜂窝的面积必须大于或者等于一个事件的最大覆盖区域；②F-分群和 S-分群必须覆盖事件所能覆盖的所有蜂窝，S-分群必须覆盖属于不同 F-分群，但又是相邻的蜂窝。

采用栅格分群有两个优点：第一个优点是，将栅格大小作为网络的一个配置参数，易于确定栅格的大小；第二个优点是，只要已知节点的地理位置，就能够立即确定节点的蜂窝、F-分群、S-分群的归属问题，其间不需要进行任何通信。地理信息在传感器网络中是必需的，所以假定传感器节点知道自己的物理位置，因此能够间接构建所有蜂窝、F-分群、S-分群。

完成栅格构建后，F-分群和 S-分群中的节点必须选择一个累积器作为自己的群首。因为作为累积器的节点的能耗多于群内其他节点，所以各个传感器节点应该轮流承担累积器，以便所有节点的能耗均匀分布。因此，必须周期性执行累积器选择进程。但是，累积器更新频率非常低，从数个小时更新一次到数天更新一次，具体取决于节点电池能量。节点可以根据某些参数（如剩余能量）按照概率选择自己作为累积器，然后通知其群内所有节点。当两个节点均选择自己作为累积器时，可以采用节点 ID 来作出决策。

另外一种方法是分群内的节点采用散列函数（Hash Function）推测一个节点在该分群内的当前时间，并使用该节点作为累积器。节点必须知道其 F-分群内所有节点的地址，并按照其节点 ID 进行归类。散列函数从 1 至 n （ n 表示其群内节点数）推测当前时间为 k ，节点使用第 k 个节点作为累积器。因为累积器变更频次很低（数小时或者数天），所以只需要粗糙时

间同步，能够避免群首选择开销。

但是，ToD 动态转发法要求每个 F-累积器知道与其 F-分群重叠的各个 S-分群的 S-累积器的位置。因此，每当 S-累积器变更时，必须通知 F-累积器。为了简化群首选择过程，避免更新信息传播开销，将 S-累积器的角色委托给 F-累积器承担。不是为一个 S-分群周期性选择一个节点作为其 S-累积器，而是为每个 S-分群选择一个 F-分群（将其称为累积分群），将这个累积分群的 F-累积器作为其 S-累积器。一个 S-分群的累积分群就是与该 S-分群重叠的所有 F-分群中离中心节点最近的那个 F-分群，如图 10-13（k）所示，中心节点位于左下角。随着 F-累积器的变化，相应的 S-累积器也随着变化。一个 F-累积器将分组转发给一个 S-累积器时，就是将分组转发给这个 S-累积器的累积分群。当分组到达累积分群时，累积分群成员节点知道其 F-累积器的位置，因此可以将分组转发给自己的 F-累积器。因此不必将累积器更新传播给相邻分群。

将 S-累积器的角色转移到 F-累积器上，S-累积器选择的 F-分群就是离中心节点最近的 F-分群。当一个 F-累积器希望将分组同时转发给两个 S-累积器时，选择离自己较近的 F-分群作为第一个 S-累积器的累积分群（可能就是本身），以便减少两个累积器之间的传输次数，如图 10-13（l）所示。这种选择不会影响分组最终会在一个累积器上被累积的性质，这是因为覆盖两个 F-分群中的蜂窝的 S-分群是相同的，因此两个 F-累积器选择的累积分群也是相同的。

上述累积器选择方法有五个优点：①不需要 S-分群的群首选择算法，因此不存在群首选择开销；②节点只需要知道其 F-分群的 F-累积器，因此是可扩展的；③当 F-累积器更改时，S-累积器也随着更改，但是这些更改不需要传播给其他 F-分群和 S-分群；④假如节点通过推测当前时间来选择累积器，获取其分群累积器的节点 ID，那么只要求相同 F-分群内节点相互时间同步；⑤因为 S-分群的累积分群是固定计算出来的，所以不存在计算累积分群的分组开销。

10.3.3 性能分析

1. 发送次数期望值

下面分析同时采用 DAA 和 RW 时网络中的总发送次数。

为了分析无结构网络中发送次数的期望值，首先计算一个分组被累积的概率。假定网络中每个节点有分组需要发送。每个节点为其产生的每个分组选择一个随机时延。假定下行节点时延大于上行节点时延，则分组能够在下行节点被累积。为了简化分析，忽略传输时延（传输时延可能会引起较少累积）。

设 Y 是一个随机变量，表示一个分组被累积前通过的转发跳数。设 $d_{vh}=x$ 表示由 v_h 选定的位于 $0\sim 1$ 之间的一个标准化随机时延， v_h 表示一个离中心节点 h 个转发跳远的节点。考虑网络中每个节点对下行节点平均有 k 个选择。若是一个分组：①经过了 $i-1$ 个节点转发，并且这 $i-1$ 个节点的时延小于发送节点的时延；②第 i 个节点的时延大于发送节点的时延，那么这个分组能够被转发至第 i 个节点，并在第 i 个节点被累积。因此，对于一个离中心节点 h 个转发跳远的节点，有

$$P(Y=i)=\begin{cases} x^{(i-1)k} \times (1-x^k), & 0 < i < h \\ x^{(i-1)k}, & i = h \end{cases} \quad (10-13)$$

当节点 v_h 的时延为 x 时, Y 的期望值为

$$\begin{aligned} E[Y | d_{vh} = x] &= \sum_{i=1}^h i \times P(Y=i) = \left[\sum_{i=1}^{h-1} i \times x^{(i-1)k} (1-x^k) \right] + h \times x^{(h-1)k} \\ &= \left(\sum_{i=1}^{h-1} i \times x^{(i-1)k} \right) - \left(\sum_{i=1}^{h-1} i \times x^{ik} \right) + h \times x^{(h-1)k} = \sum_{i=0}^{h-1} x^{ik} \end{aligned} \quad (10-14)$$

因为 x 位于 $0 \sim 1$ 之间, 所以期望值 $E[Y]$ 为

$$E[Y] = \int_0^1 E[Y | d_{vh} = x] dx = \int_0^1 \left(\sum_{i=0}^{h-1} x^{ik} \right) dx = \left[\sum_{i=0}^{h-1} \frac{x^{ik+1}}{ik+1} \right]_0^1 = \sum_{i=0}^{h-1} \frac{1}{ik+1} \quad (10-15)$$

运用式 (10-19) 就能够计算出网络中发送次数的期望值, 即

$$\sum_{h=1}^{n/k} k \sum_{i=0}^{h-1} \frac{1}{ik+1} = (n+1)H_k \left(\frac{n}{k} \right) - \frac{n}{k} \quad (10-16)$$

式中, $H_k(n) = \sum_{i=1}^n \frac{1}{(i-1)k+1}$ 是调和级数之和。

下面以链式拓扑网络为例作进一步分析。

考虑一个从节点 v_0 到节点 v_n 的链式拓扑, 如图 10-14 (a) 所示, 其中节点 v_0 是中心节点, 所有节点有数据需要报告给中心节点 (v_0)。每个节点从 $0 \sim 1$ 中随机选择一个数, 等效于选择发送顺序的随机排列。节点 v_n 产生的分组需要转发 h 个转发跳, 因此节点 v_n 必须在 $h-1$ 跳范围内的所有节点发送完之后再发送, 但是必须在第 h 跳节点之前发送。这等效于给这 n 个节点随机分配 n 个不同的数 (s_{v_1} 到 s_{v_n}) 作为其发送顺序, 使得 $s_{v_{n-h}}$ 在节点 v_{n-h} 到 v_n 之间是最大数, s_{v_n} 是第二最大数。从 n 个数中选择 $h+1$ 个数, 有 $\binom{n}{h+1}$ 种选择。在所选出的 $h+1$ 个数中, 有 $(h-1)!$ 种排列顺序, $s_{v_{n-h}}$ 是最大数, s_{v_n} 是第二最大数。剩余的 $[n-(h+1)]$ 个数有 $[n-(h+1)]!$ 种排列。因此, 一个分组被转发 h 个转发跳的概率为 $\left[\binom{n}{h+1} \times (h-1)! \times (n-(h+1))! \right] / n! = 1/[h(h+1)]$ 。

假如分组没有被累积而传递到达中心节点, 那么其传输路由上所有节点的时延必须小于源节点的时延。对于 n 个节点, 排列分数是 $1/n$ (其中一个给定节点具有最大时延)。因此

$$P(Y=i)=\begin{cases} \frac{1}{i(i+1)}, & 0 < i < h \\ \frac{1}{n}, & i = h \end{cases} \quad (10-17)$$

节点 v_n 的期望值 $E[Y]$ 为

$$E[Y] = \sum_{i=1}^n i \times P(Y=i) = \sum_{i=1}^{n-1} i \times \frac{1}{i(i+1)} + n \times \frac{1}{n} = \sum_{i=1}^n \frac{1}{i} \quad (10-18)$$

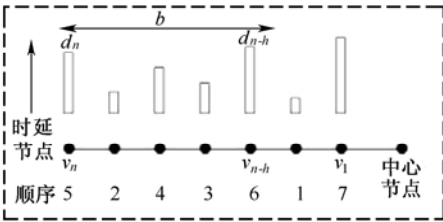
因此，链式拓扑网络中发送次数的期望值为

$$\sum_{h=1}^n \sum_{i=1}^h \frac{1}{i} = (n+1)H_1(n) - n = (n+1)[\ln n + O(1)] - n \approx n \ln n, \quad n \rightarrow \infty \quad (10-19)$$

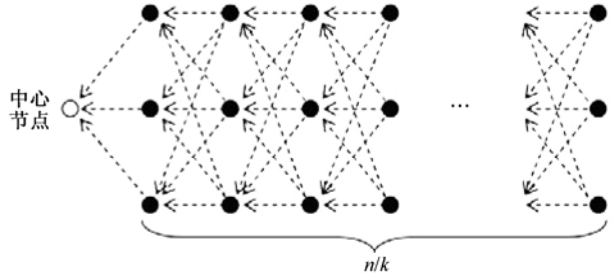
当 $k=1$ 时，式 (10-19) 符合式 (10-16)，因此证明前面的分析正确。

使用图 10-14 (b) 所示的网络拓扑比较分析结果与 ns-2 仿真结果。每个节点有 3 个下行节点处在其传输覆盖范围内。在仿真中，节点只将分组发送给下一列 3 个下行节点中的一个，而不会将分组发送给同一列的节点以及远离中心节点的节点。分组发送的最大延迟时间 $\tau=50$ ，单位是单个最大分组的传输时间，其中包括 50 B 分组的任意组播开销 (约 0.04 s)。因此，最大随机等待时延为 2 s。

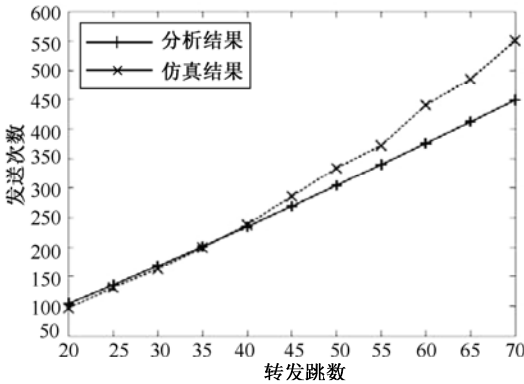
图 10-14 (c) 给出了发送次数的仿真结果与分析结果。从图 10-14 (c) 中可以看到：当网络规模小于 40 个转发跳时，分析结果符合仿真结果；当转发跳数继续增大时，仿真的发送次数增大比分析结果快，其原因在于分析时没有使用传输时延模型，而传输时延不能忽略，所以不会在时延较大的下行节点上累积分组，这个效果随着转发跳数的增大而增强，因此仿真结果与分析结果的差异也随着增大。 τ 越大，仿真结果接近分析结果的网络规模越大。尽管只是在源节点引入时延，但是 τ 太大时得到的端到端时延是不能接受的。



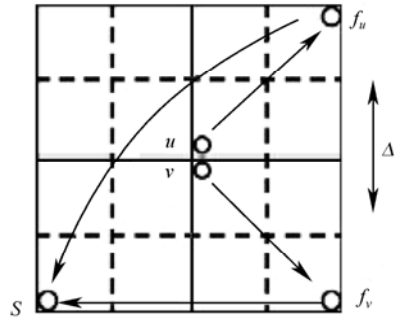
(a) 节点随机选择时延时的发送顺序



(b) 用于比较的网络拓扑 (每个节点有 $k=3$ 个下行节点选择)



(c) 发送次数与网络规模的变化关系
(仿真结果与分析结果的对比, $\tau=50$)



(d) ToD 的最差情形

图 10-14 DAA、RW、ToD 的性能分析

2. ToD 分析

下面说明 ToD 上任意两个相邻节点之间的最大距离只依赖蜂窝的大小，而与网络大小无

关。因为进行完全累积（即所有分组被累积成一个分组且分组不会增大），经过累积后只有一个分组从累积器转发给中心节点，所以开销相对较小，因而忽略从累积器到中心节点的开销。采用固定累积树结构的栅格网络的下限是 $O(\sqrt{n})^{[21]}$ ，而 ToD 即使在最差情形下也仍然能够达到恒定因数。

图 10-14（d）表示的 ToD 最差情形是：在两个不同 F-分群的角落中分别只有一个节点 u 和 v 产生分组， u 和 v 是相邻节点，两个 F-累积器 f_u 和 f_v 分别位于节点 u 和 v 的对角的另一端。假定传感器节点密集布置，因此将两个传感器节点之间的距离转换为之间发送一个分组的开销。图 10-14（d）是最差情形，这是因为：即使一个分群中有更多节点产生分组，也只是分摊从 F-累积器到 S-累积器的分组发送开销，多个 F-分群中更多节点产生分组只会降低平均距离。

假定蜂窝边长为 Δ 。假定若两个节点之间的距离小于一个单位距离，则这两个节点是相邻节点。因此在图 10-14（d）中， u 和 v 产生的分组在 s 被累积前所转发的距离等于 u 到达 f_u 的距离、 f_u 到达 s 的距离、 v 到达 f_v 的距离、 f_v 到达 s 的距离之和，即等于 $(2\Delta\sqrt{2} + 4\Delta\sqrt{2}) + (2\Delta\sqrt{2} + 4\Delta) = 8\Delta\sqrt{2} + 4\Delta$ 。因此，若采用最佳方法，那么由于 u 和 v 是相邻节点，所以只需要一次发送。在 ToD 中，对于图 10-14（d）所示最差情形，需要 $(8\Delta\sqrt{2} + 4\Delta)$ 次发送。

但是，由于采用 DAA 技术，所以来自相邻节点的分组被立即累积。因此，要发生图 10-14（d）所示的最差情形， u 和 v 之间的距离至少必须等于 2 个距离单位，ToD 动态转发协议的发送次数是最佳发送次数的 $(4\Delta\sqrt{2} + 2\Delta)$ 倍，约等于 7.66Δ 。这个上限只依赖蜂窝的大小，而蜂窝的大小依赖事件覆盖范围的大小。这个上限与网络大小无关，因此非常适合于大规模网络。

平均发送次数比 $4\Delta\sqrt{2} + 2\Delta$ 小得多，这是因为：①通常有许多节点产生分组；②一个节点与其 F-累积器之间的距离并不总是等于 $2\Delta\sqrt{2}$ ，F-累积器与 S-累积器之间的距离也比较短；③DAA 法能够有效累积相邻节点的分组，从而进一步减少发送次数。

10.3.4 ToD和DAA的性能

利用 ns-2 网络仿真器在大规模传感器网络下评估和比较 ToD、DAA、SPT、OPT 四个协议的性能和扩展性，其中 OPT 为最佳累积树（Optimal Aggregation Tree）协议，节点在一棵累积树上转发其分组，树根在事件的中心位置；节点知道分组转发至何处以及等待时间长度；假定已知事件的位置和移动性，事先构建的 OPT 树随着事件移动而变化；理想情况下，对于 n 个源节点只需要 $n-1$ 次发送，这是任何累积结构的下限值，因此将其作为最佳情形。

仿真区域为 $2\,000\text{ m} \times 1\,200\text{ m}$ 的栅格网络，节点间距 35 m ，总共布置 $1\,938$ 个节点，电台数据速率 38.4 kb/s ，电台传输距离稍高于 50 m ，事件按照随机点移动模型在网络中移动、移动速度 10 m/s 、持续 400 s ，事件覆盖区域直径 400 m 。一个事件触发的节点按周期 5 s 位于 $(0,0)$ 的中心节点发送分组，采用完全累积。

比较的性能主要是标准化发送次数。标准化发送次数指整个网络将一个单位有用信息从源节点发送至中心节点所执行的平均发送次数。假如已知一次发送的发送开销和接收开销，那么就能够将标准化发送次数转换为标准化能耗，因此可以推导出收集一个分组数据的能耗。在所有测试床实验和仿真实验中，对于所比较的所有累积协议，所有节点是完全活动的，空

闲能耗大致相同，所以不考虑空闲能耗。

1. 事件大小

首先评估产生分组的节点数对这些协议的性能影响。这些仿真实验反映事件大小对每种协议的性能影响。

图 10-15 (a) 表示标准化发送次数的结果。ToD 的性能分别比 DAA、SPT 提高了 30%、85%，比 OPT 提高了 25%。但是当事件移动时采用连续变化的累积树时，OPT 表现最好（仿真中未考虑事件移动的开销）。SPT 是概率累积，所以表现最差。除了 SPT，其他三种协议表现相当平稳，这就说明这三种协议具有相当良好的事件大小可扩展性。

图 10-15 (b) 表示总发送次数，图 10-15 (c) 表示中心节点接收到的有用信息总量。从图 10-15 中可以看到：DAA 和 ToD 的接收分组多于 OPT，这是因为 DAA 和 ToD 采用的无结构累积方法能够在分组产生早期就对其进行累积，并将分组相互分散开以减少竞争；ToD 的标准化发送次数优于 DAA，这是因为 ToD 能够在接近源节点的节点累积分组，因此降低了从源节点至中心节点的分组转发开销。

2. 可扩展性

假如一个协议不是可扩展的，那么其性能将随着网络规模的增大而下降。为了评估 ToD、DAA、SPT、OPT 四个协议的可扩展性，假定事件只在离中心节点一定距离的有界区域内移动。因此，限制事件只在一个 400 m×1 200 m 矩形内移动，矩形离中心节点的距离可变范围为 200~1 400 m。为了公平比较，事件移动不会接近 200 m 的网络边界，以确保事件触发的节点数不会发生剧烈变化。实验结果如图 10-15 (d) ~图 10-15 (f) 所示，从图中可以看到：ToD 和 OPT 的性能保持平稳，ToD 的性能比 OPT 提高 22%。ToD 的性能不会随着网络规模的增大而下降，这就说明 ToD 的可扩展性相当不错。DAA 和 SPT 的性能随着网络规模的增大而下降。当事件从最近矩形（离中心节点）移动到最远矩形时，DAA 和 SPT 的标准化发送次数增加一倍。

图 10-15 (f) 表示中心节点关于每个事件的接收分组数量。假如在事件附近累积所有分组后再将其转发给中心节点，那么中心节点关于一个事件将只接收一个分组。反之，中心节点接收越多分组则表明网络中进行的分组累积越少。随着网络规模的增大，给中心节点转发越多分组的开销将迅速提高。从图 10-15 (f) 中看到：对于 DAA 和 SPT，中心节点接收很多分组；尽管中心节点接收的分组数量保持相当平稳，但是总发送次数随着从源节点至中心节点的距离的增大而线性递增。

理想情况下，假如所有分组被累积器所累积，那么中心节点关于一个事件将只接收一个分组。但是对于 ToD 和 OPT，中心节点关于一个事件的接收分组数量大于 1，这是因为无法精确预测基于 CSMA 的 MAC 协议中的时延，因此累积器可能在接收到所有被转发来的分组之前就将分组转发给中心节点。在 ToD 和 OPT 中，尽管累积器将未累积分组转发给中心节点的开销也随着网络规模增大而提高，但是提高幅度相对小于 DAA 和 SPT，这是因为未经累积就被转发给中心节点的分组极少。

从图 10-15 (f) 中观察到：对于 ToD，当事件接近中心节点时，中心节点所收分组较多。在仿真中，假定中心节点通过有线供电，因而不需要将累积器角色委托给其他节点，实现能

耗的均匀分布，因此相同 F-分群中作为中心节点的节点总是选择中心节点作为 F-累积器。

3. 蜂窝大小

上述仿真实验采用最大事件覆盖范围作为蜂窝的大小，确保动态转发能够在 S-累积器累积所有分组，将累积结果分组转发给中心节点的开销达到最低。但是，由于在 F-分群中采用 DAA 累积技术，所以采用大蜂窝导致累积器分组累积开销增大。下面介绍蜂窝大小对 ToD 的性能影响。

蜂窝变化范围为 $50\text{ m}\times 50\text{ m}\sim 800\text{ m}\times 800\text{ m}$ ，取三种蜂窝直径 200 m、400 m、600 m 进行仿真实验。从 5 种不同事件移动模式中收集的仿真结果如图 10-15 (g)～图 10-15 (i) 所示。

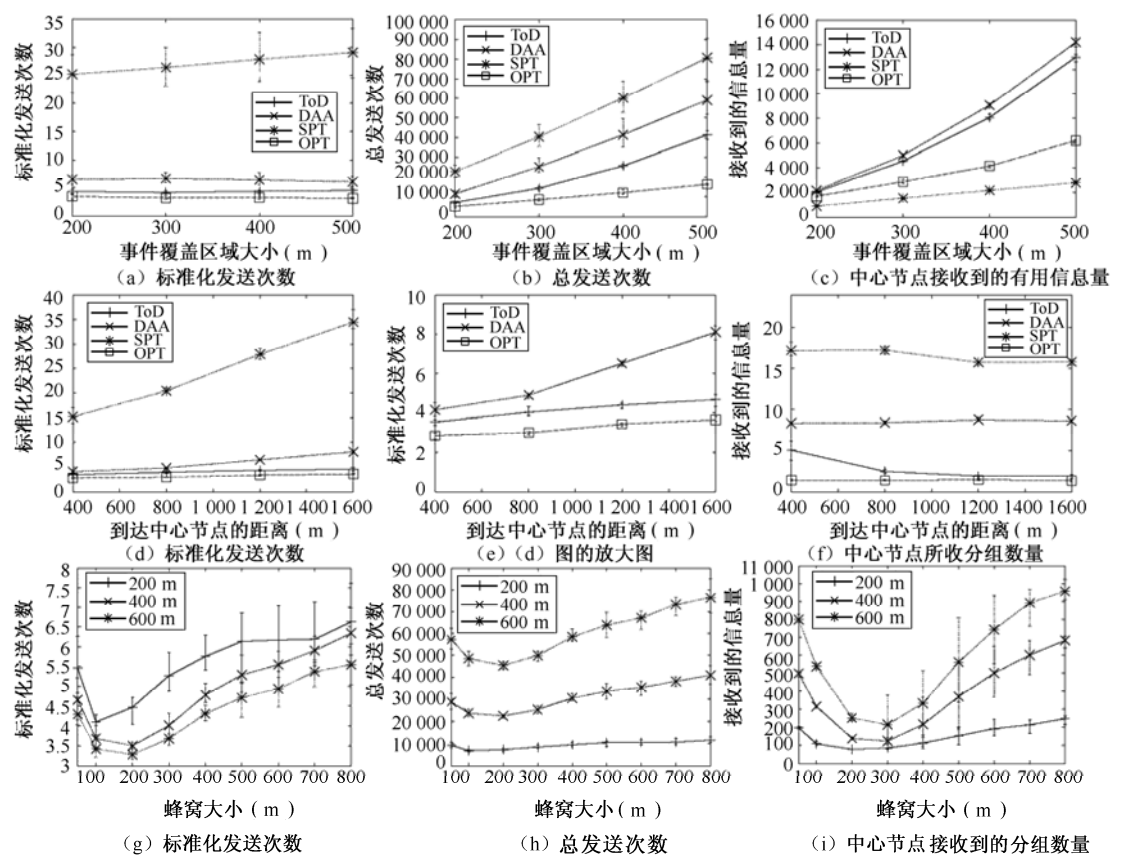


图 10-15 ToD 和 DAA 的仿真结果

当蜂窝大于事件覆盖范围时，性能变差，这是因为 F-累积器的分组累积开销增大，但是从 S-累积器至中心节点的分组转发开销保持不变。当蜂窝太小时，由于分组在不同 F-累积器被累积，较多分组未经进一步累积就被转发给中心节点，所以对中心节点的分组转发开销增大。一般来讲，当 F-分群小到只包含一个节点或者当 F-分群大到包含所有网络节点时，ToD 正好降为 DAA。

当蜂窝大小为 $100\text{ m}\times 100\text{ m}$ (F-分群大小为 $200\text{ m}\times 200\text{ m}$)、事件覆盖区域直径为 200 m

时, ToD 性能最佳。当事件覆盖区域直径为 400 m 和 600 m 时, 采用 200 m×200 m 的蜂窝大小, F-分群大小为 400 m×400 m, ToD 性能最好。因此, 可以根据蜂窝大小进一步优化 ToD 性能。

参 考 文 献

- [1] M. Ding, X. Cheng, G. Xue. Aggregation tree construction in sensor networks. in: Proc. of IEEE VTC'03, Orlando, FL, Oct. 2003, pp.2168–2172.
- [2] M. Lee, V.W.S. Wong. An energy-efficient spanning tree algorithm for data aggregation in wireless sensor networks. in: Proc. of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PacRim), Victoria, BC, Aug. 2005.
- [3] M. Lee, V.W.S. Wong. LPT for data aggregation in wireless sensor networks. in: Proc. of IEEE Globecom'05, St. Louis, Missouri, Nov./Dec. 2005.
- [4] M. Lee, V.W.S. Wong. E-Span and LPT for data aggregation in wireless sensor networks. *Computer Communications* 29 (2006) 2506-2520.
- [5] T. He, B. M. Blum, J. A. Stankovic, and T. Abdelzaher. Aida: Adaptive Application-independent Data Aggregation in Wireless Sensor Networks. *ACM Transactions on Embedded Computing Systems*, 3(2):426-457, May 2004.
- [6] C. Intanagonwiwat, D. Estrin, and R. Goviindan. Impact of Network Density on Data Aggregation in Wireless Sensor Networks. in Technical Report 01-750, University of Southern California, November 2001.
- [7] W. Zhang and G. Cao. Optimizing Tree Reconfiguration for Mobile Target Tracking in Sensor Networks. in Proceedings of INFOCOM 2004, Vol.4, March 2004, pp.2434–2445.
- [8] W. Zhang and G. Cao. DCTC: Dynamic Convoy Tree-based Collaboration for Target Tracking in Sensor Networks. in *IEEE Transactions on Wireless Communications*, Vol.3, September 2004, pp.1689–1701.
- [9] E. L. Lawler, J. K. Lenstra, A. H. G. R. Kan, and D. B. Shmoys, *The Traveling Salesman Problem : A Guided Tour of Combinatorial Optimization*. JohnWiley & Sons, 1985.
- [10] R. Cristescu, B. Beferull-Lozano, and M. Vetterli. On Network Correlated Data Gathering. in Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies, Vol.4, March 2004, pp.2571–2582.
- [11] K. W. Fan, S. Liu, and P. Sinha. Scalable Data Aggregation for Dynamic Events in Sensor Networks. *SenSys'06*, pp.181–194, November 2006.
- [12] K. W. Fan, S. Liu, and P. Sinha. Structure-free Data Aggregation in Sensor Networks. in OSU-CISRC-4/06-TR35, Technical Report, Dept of CSE, OSU, April 2006.
- [13] K. W. Fan, S. Liu, and P. Sinha. On the potential of Structure-free Data Aggregation in Sensor Networks. in To be appear in Proceedings of INFOCOM 2006, April 2006.
- [14] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks. in Proceedings of the 5th symposium on Operating systems design and implementation, December 2002, pp.131–146.

- [15] S. Madden, R. Szewczyk, M. J. Franklin, and D. Culler. Supporting Aggregate Queries Over Ad-Hoc Wireless Sensor Networks. in Proceedings of the 4th IEEE Workshop on Mobile Computing Systems and Applications, June 2004, pp.49–58.
- [16] S. Lindsey, C. S. Raghavendra, and K. M. Sivalingam. Data Gathering in Sensor Networks using the Energy Delay Metric. in Proceedings 15th International Parallel and Distributed Processing Symposium, April 2001, pp.2001–2008.
- [17] S. Lindsey, C. Raghavendra, and K. M. Sivalingam. Data Gathering Algorithms in Sensor Networks Using Energy Metrics. in IEEE Transactions on Parallel and Distributed Systems, Vol.13, September 2002, pp.924–935.
- [18] R. Cristescu and M. Vetterli. Power Efficient Gathering of Correlated Data: Optimization, NP-Completeness and Heuristics. in Summaries of MobiHoc 2003 posters, Vol.7, July 2003, pp.31–32.
- [19] S. Pattern, B. Krishnamachari, and R. Govindan. The Impact of Spatial Correlation on Routing with Compression in Wireless Sensor Networks. in Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks, April 2004, pp.28–35.
- [20] L. Cai and D. Corneil. Tree Spanners. in SIAM Journal of Discrete Mathematics, Vol.8, 1995.

第 11 章 无线传感器网络安全

对于许多传感器网络应用，安全是非常关键的。有些传感器网络应用不仅要面对苛刻的环境，而且还要面对主动、智能的对手，因此战场上的传感器网络需要具有抗定位、破坏、颠覆的能力。在其他场合，安全需求虽然不明显，但确实需要。

① 灾难：对于人员伤亡的地点和情况，特别是对于跟正在进行的恐怖分子活动有关的不幸事件（而不是自然灾害），可能有必要预防未得到授权而进行的揭发和报道。

② 公共安全：有关化学、生物、环境威胁的虚假警报可能会引起惊慌，甚至对报警系统的冷漠。对这种系统的攻击可能先于对手保护资源的真正攻击。

③ 家庭健康：因为保护秘密非常重要，所以只有获得授权的用户才能够查询或者监视传感器网络。传感器网络也可以构成事故-通知链的主要数据，因此必须受到保护，不能出现故障，甚至失效。

协议、软件在开始设计时就应该考虑其安全，特别是关于抗网络有效性攻击的安全是必须考虑的。以后试图增加安全功能不方便，且经常是不成功的。

11.1 WSN安全概述

11.1.1 WSN安全威胁模型

在 WSN 中，通常假定攻击者可能知道传感器网络中使用的安全机制，能够危及某个传感器节点的安全，甚至能够捕获某个传感器节点。由于布置具有抗篡改能力的传感器节点成本高，所以认为大多数 WSN 节点没有抗篡改能力。一旦一个节点存在安全威胁，那么攻击者可以窃取这个节点内的密钥。WSN 中的中心节点通常认为是可信的。

传感器网络的攻击分成以下几类：

① 外部攻击与内部攻击：外部攻击定义为来自本 WSN 之外的节点的攻击；当 WSN 的合法节点进行无意识操作或者未授权操作时，即发生内部攻击。

② 被动攻击与主动攻击：被动攻击包括偷听、监视 WSN 内交换的分组；主动攻击涉及对数据流的某种程度修改、创建虚假数据流。

③ 传感器类攻击与微型计算机类攻击：在传感器类攻击中，攻击者使用少数几个与 WSN 网络节点能力类似的节点攻击这个 WSN；在微型计算机类攻击中，攻击者采用较强装置（比如微型计算机）攻击 WSN，这种攻击装置传输距离更远、处理能力更强、存储能量更多（相对于 WSN 网络节点）。

11.1.2 WSN安全面临的障碍

WSN 是一种特殊类型的网络，其约束条件很多（相对于传统计算机网络）。这些约束条

件导致很难将现有的安全技术应用到 WSN 中。下面分析 WSN 的约束条件。

1. WSN资源极其有限

所有的安全协议和安全技术都需要依靠一定资源来实现，包括数据存储器、程序代码存储器、能量以及带宽。但是，目前无线微型传感器中的这些资源极其有限。

(1) 存储器容量限制

传感器节点是微型装置，只有少量存储器用于存储代码。为了建立有效安全机制，有必要限制安全算法的实现代码长度。例如一个 Mica 传感器节点只有 128 KB 的代码存储容量，4 KB 的数据存储容量。TinyOS 代码约占 4 KB。因此，所有安全实现代码必须很小。

(2) 能量限制

能量是无线传感器能力的最大约束因素。通常依靠电池供电的传感器节点一旦布置在一个传感器网络中后就不容易被替换（工作成本很高），也不容易重新充电（传感器成本高），因此必须节省电池能量，延长各个传感器节点的寿命，从而延长整个传感器网络的寿命。在传感器节点上实现一个加密函数或者协议时，必须考虑所增加的安全代码对能量的影响。给传感器节点增加安全能力时，必须考虑这种安全能力对节点寿命（即电池工作寿命）的影响。节点安全能力引起的能耗包括所要求的安全功能（如加密、解密、数据签名、签名验证）的处理能耗、有关安全数据和开销（如加密/解密所需要的初始化矢量）的发送能耗、采用安全方式存储安全参数的能耗（如加密密钥的存储）。

2. 不可靠通信

不可靠通信无疑是 WSN 安全的另一个威胁。WSN 安全密切依赖所定义的协议，而协议又依赖通信。

(1) 不可靠传输

传感器网络的分组传输路由是无连接路由的，因此不可靠。信道误码、高拥塞节点的分组丢失可能损坏分组，结果导致分组丢失。不可靠的无线通信信道也会损坏分组。高信道误码率迫使软件开发人员利用一些网络资源来处理误码。假如协议没有合适的误码处理能力，那么有可能丢失关键的安全分组（如加密密钥）。

(2) 碰撞

即使信道可靠，通信也仍然可能不可靠，其原因在于 WSN 的广播特性。假如分组在传输途中遇到碰撞，那么分组传输失败。在高密度传感器网络中，碰撞是一个主要问题。

(3) 时延

多跳路由、网络拥塞、节点处理会引起较大的网络时延，因此实现传感器节点之间的同步很困难。同步问题对传感器安全很关键：安全机制依赖关键事件报告和加密密钥分组。

3. WSN网络操作无人照看

依据具体传感器网络的特定功能，传感器节点可能长时间处于无人照看状态。对于无人照看传感器节点存在以下三个主要威胁：

① 暴露在物理攻击之下。传感器节点可能布置在对攻击者开放、恶劣气候等环境中。这种环境中的传感器节点遭受物理攻击的可能性比典型 PC（安置在一个安全地点，主要面临

来自网络攻击)要高得多。

② 远程管理。传感器网络的远程管理实质上不可能检测出物理篡改、进行物理维护(如替换电池)。最典型的例子是用于远程侦查的传感器节点(布置在敌方边界之后)可能失去与友方部队的联系。

③ 缺乏中心管理点。一个 WSN 应该是一个分布式网络,没有中心管理点,这会提高 WSN 的生命力。但是,假如设计不合理,会导致网络组织困难、低效、脆弱。

传感器节点无人照看时间越长,受到攻击者安全攻击的可能性就越大。

11.1.3 WSN安全要求

WSN 安全服务的目标就是防止信息和网络资源受到攻击和发生异常。

1. 数据机密性

数据机密性是网络安全中最重要的内容。每个网络的任何安全重点通常首先就是解决数据机密性问题。在 WSN 中,一个传感器网络不应该将其传感器感知数据泄漏给邻近网络,特别是在军事应用中,传感器节点存储的数据可能高度敏感。在很多 WSN 应用中(如密钥分发),节点发送高度敏感数据,因此在 WSN 中建立安全信道尤其重要。公用传感器信息(如传感器节点身份识别码 ID、公共密钥等)也应该被加密,在一定程度上防止流量分析攻击。

保持敏感数据秘密的标准方法是采用秘密密钥加密敏感数据,只有预定接收节点才有秘密密钥,因此可以实现机密性。对于给定通信模式建立节点与中心节点之间的安全信道以及独立完成随后必需的其他安全信道。

2. 数据完整性

实现数据机密性后,攻击者不能窃取信息,但是并不意味着数据就是安全的。攻击者能够修改数据,使 WSN 进入混乱状态。例如,恶意节点可以在分组中添加一些数据分片或者篡改分组中的数据,然后将改变后的分组发送给原始接收节点。即使不存在恶意节点,但由于通信环境条件恶劣,所以仍然会发生数据丢失或者数据受损。因此在通信中,数据完整性确保接收节点所接收数据在传输途中不会被攻击者篡改。SPIN 采用数据认证来实现数据完整性。

3. 数据新鲜度

即使能够保证数据机密性和数据完整性,但是仍然必须确保每条消息的新鲜度。数据新鲜度意味着数据是最近的,确保不是攻击者重放的旧消息。当采用共享密钥策略时,这个要求尤其重要,通常共享密钥必须随时改变。但是,将新的共享密钥传播给整个网络需要一定时间。此时,攻击者很容易进行重放攻击。假如传感器节点意识不到随时改变新密钥,那么很容易破坏传感器节点的正常工作。为了解决这个问题,可以在分组中添加一个随机数或者跟时间有关的计数器,确保数据新鲜度。

SPIN 识别两种类型的新鲜度:弱新鲜度——提供局部消息排序,但是不承载时延信息;

强新鲜度—提供全部请求—响应对的排序，允许时延估计。弱新鲜度用于传感器感知数据，强新鲜度用于网内时间同步。

4. 认证

消息认证对很多传感器网络应用（例如网络重新编程、控制传感器节点占空因数之类的管理任务）都非常重要。攻击者并不局限于修改数据分组，还能够通过注入额外分组而改变整个分组流，所以接收节点必须确保决策过程中使用的数据来自正确的可信任源节点。接收节点通过数据认证验证数据确实是所要求的发送节点发送的。

对于点对点通信，可以采用完全对称机制实现数据认证。发送节点和接收节点共享一个秘密密钥，秘密密钥用于计算所有通信数据的消息认证码（Message Authentication Code, MAC）。接收节点接收到一条具有正确消息认证码的消息时，就知道这条消息必定是与其通信的那个合法发送节点发送的。

在广播环境中不能对网络节点作出较高的信任假设，因此这种认证技术不适用于广播环境。假如一个发送节点需要给互不信任的接收节点发送消息，那么使用一个对称消息认证码是不安全的：其中任何一个不信任接收节点只要知道这个对称消息认证码，就可以扮演成这个发送节点，伪造发送给其他接收节点的消息。因此，需要非对称机制来实现广播认证。

5. 可用性

调整、修改传统加密算法而使其适用于 WSN 不方便，而且会引入额外开销。或者修改代码，使其尽可能重复使用；或者采用额外通信实现相同目标；或者强行限制数据访问，这些方法都会弱化传感器和传感器网络的可用性，理由如下：

- 额外计算消耗额外能量，若不再有能量，数据则不再可用；
- 额外通信也消耗较多能量，而且，通信增加，通信碰撞概率随着增大；
- 假如使用中心控制方案，那么会发生单点失效问题，由此极大地威胁网络可用性。

可用性安全要求不仅影响网络操作，而且对于维护整个网络的可用性非常重要。可用性确保：即使存在 QoS 攻击，所需网络服务仍然可用。

6. 自组织

WSN 一般是 Ad Hoc 网络，要求每个传感器节点具有足够的独立性和灵活性，能够按照不同情况进行自组织、自愈。网络中不存在固定基础设施，用于网络管理。这个固有特征给 WSN 安全带来一个极大的挑战。例如，整个网络的动态性导致无法预先配置中心节点与所有传感器节点共享的密钥。于是提出了若干种随机密钥预分配方案。若在传感器网络中采用公共密钥加密技术，则必须具有公共密钥高效分发机制。分布式传感器网络必须能够自组织，支持多跳路由和密钥管理，建立传感器节点之间的信任。假如传感器网络自组织能力不足或者缺乏自组织能力，那么攻击者甚至危险环境造成的网络受损都可能是毁灭性的。

7. 时间同步

大多数传感器网络应用依靠某种形式的时间同步。为了节省能量，各个传感器定期关闭

其电台。传感器节点需要计算分组在两个通信节点对之间的端到端时延。联合协作性传感器网络用于跟踪应用时，可能需要节点组同步。

8. 安全定位

一个传感器网络的效用常常依赖于每个网络节点精确而自动的节点定位能力。故障定位传感器网络需要精确的位置信息才能够查明故障的位置。但是，攻击者很容易操控不安全的位置信息，如报告虚假信号强度和重放信号等。

9. 其他安全要求

授权：授权确保只有得到授权的传感器节点才能够参与对网络服务的信息提供。

认可：认可表示节点不能拒绝发送其以前已经发送过的消息。

在 WSN 中，在网络运行过程中发生传感器节点失效问题、布置新的传感器节点是很常见的，因此应该考虑前向保密要求和后向保密要求：

- 前向保密：一个传感器节点退网后应该不能再读取网络中随后的任何消息；
- 后向保密：入网节点应该不能读取网络中此前已经发送过的任何消息。

11.1.4 WSN安全解决方案的评估

使用如下一些性能指标和能力来评估一个 WSN 安全解决方案是否适合 WSN：

- 安全：安全解决方案必须满足 WSN 安全要求；
- 弹性：当少数几个节点存在安全威胁时，安全解决方案应该能够继续防止攻击；
- 能量效率：安全解决方案必须是能量高效的，才能够达到最大的节点寿命和网络寿命；
- 灵活性：要求密钥管理灵活，适用于各种不同的网络布置方法，如随机的节点扩散、预先确定的节点布置；
- 可扩展性：安全解决方案具有可扩展能力，不会对安全要求造成不利影响；
- 容错能力：在发生故障（如节点失效）时，安全解决方案应该继续提供安全服务；
- 自愈能力：传感器节点可能失效或者耗尽其能量，剩余传感器节点可能需要重组，继续维持一定程度的安全；
- 保证（Assurance）：保证是按照不同安全等级给端用户分发信息的能力，安全解决方案应该提供有关所需的可靠性、时延等选择。

11.2 WSN中的安全攻击

WSN 易受各种攻击。根据 WSN 的安全要求，对 WSN 的攻击归类如下：

- 对秘密和认证的攻击：标准加密技术能够保护通信信道的秘密和认证，使其免受外部攻击（比如偷听、分组重放攻击、分组篡改、分组哄骗）；
- 对网络有效性的攻击：对网络有效性的攻击常常称为拒绝服务（Denial of Service, DoS）攻击，可以针对传感器网络任意协议层进行 DoS 攻击；

- 对服务完整性的秘密攻击：在秘密攻击中，攻击者的目的是使传感器网络接收虚假数据，例如攻击者威胁一个传感器节点的安全，并通过这个节点向网络注入虚假数据。

在这些攻击中，使传感器网络继续发挥其预定作用是必要的。DoS 攻击通常就是攻击者针对网络进行的破坏、扰乱、毁灭。一种 DoS 攻击可以是削弱或者消除网络执行其预定功能的能力的任何事件。由于能够针对传感器网络任意协议层进行 DoS 攻击，所以层次化体系结构使得 WSN 在面对 DoS 攻击时很脆弱。下面按照 WSN 协议层次结构分析 WSN 的安全攻击。

11.2.1 物理层安全攻击

物理层负责频率选择、载波频率生成、信号检测、调制/解调、数据加密/解密。传感器网络是 Ad Hoc 大规模网络，主要采用无线通信，无线传输媒介是开放式媒介，因此在 WSN 中有可能存在人为干扰。对于布置在敌方环境或者不安全环境中的 WSN 节点，攻击者很容易进行物理访问。

1. 人为干扰

对无线通信的一种众所周知的攻击就是采用干扰台干扰网络节点的工作频率。一个干扰源只要功率足够大，能够破坏整个 WSN；如果功率比较低，只能破坏网络中的一个较小区域。即使采用功率较低的干扰源，假如干扰源随机分布在网络中，那么攻击者仍然有可能破坏整个网络。攻击者使用 k 个随机分布的干扰节点就能够破坏整个网络，使 N 个节点处于服务之外， k 比 N 小得多。对于单个频率的网络，这种攻击既简单又有效。

抗人为干扰的典型技术就是采用各种扩频通信技术（如跳频、码扩）。跳频扩频（Frequency Hopping Spread Spectrum, FHSS）就是发送信号时使用发射机和接收机均知道的伪随机序列在许多频率之间迅速切换载波频率。攻击者若不能跟踪频率选择序列，则不能及时干扰给定时刻的工作频率。但是，由于工作频率范围是有限的，所以攻击者可以干扰工作频带的很大一部分甚至整个工作频带。

码扩是用来对抗人为干扰的另一种技术，通常用于移动网络中。码扩设计复杂性较高，能量需求也较高，从而限制了其在 WSN 中的应用。一般地，为了维护低成本和低功耗要求，传感器装置采用单频率工作，因此极易受人工干扰攻击。

假如攻击者持久性采用干扰台干扰整个网络，那么就会得到有效而完整的 DoS 效果。因此，传感器节点应该具有对抗人工干扰的策略，比如切换到较低占空因数，尽量节省能量。节点周期性苏醒，检查人工干扰是否已经结束。传感器节点通过节省能量可能能够承受得住攻击者的人工干扰，此后攻击者必须以更高的成本进行人工干扰。

假如人工干扰是断断续续的干扰，那么传感器节点可以采用高功率给中心节点发送几条高优先级的消息，将人工干扰报告给中心节点。各个传感器节点应该相互协作，共同努力将这些消息交付给中心节点。传感器节点也可以不定期地缓存高优先级消息，等待在人工干扰间隙将其中继给其他传感器节点。

对于大规模 WSN，攻击者要成功干扰整个网络比较困难；假如进行干扰的只是被攻击者攻克的原网络节点，那么要成功干扰整个网络就更加困难。

2. 物理篡改

攻击者也可以从物理上篡改 WSN 节点、询问和危害 WSN 节点，这些是导致大规模、Ad Hoc、普遍性的 WSN 不断恶化的安全威胁。实际上，实施对分布在数千米范围内的几百个传感器节点的访问控制是极困难的，甚至是不可能的。WSN 不仅要承受武力破坏，而且还要承受较复杂的分析攻击。攻击者可以毁坏 WSN 节点，使其丧失正常工作能力；替换 WSN 节点中的关键组件（如传感器硬件、计算硬件，甚至软件），将 WSN 节点变成失密节点，从而对其实现掌控；也可以提取 WSN 节点中的敏感组件（如加密密钥），以便能够自由访问高层通信。可能无法区分节点被毁、节点故障静默这两种情形。

物理篡改的一种对抗措施是篡改验证节点的物理层分组。这种对抗措施的成功依赖于：

① WSN 设计者在设计 WSN 时就精确、完整地考虑可能存在的物理安全威胁；② 可用于设计、结构、测试的有效资源；③ 攻击者的智慧高低和果断程度。但是，这种对抗措施通常假定在 WSN 中，由于额外的成本开销，传感器节点是不能篡改验证的。这就意味着安全机制必须考虑传感器节点被危害的情形。

11.2.2 链路层安全攻击

MAC 层为相邻节点到相邻节点的通信提供信道仲裁。基于载波侦听的协作性 MAC 协议特别易受 DoS 攻击。

1. 碰撞

攻击者只需要发送一个字节就可能产生碰撞，从而损坏整个分组。分组中的数据部分发生变化，则在接收方不能通过校验和检验。ACK 控制消息被损坏会引起有些 MAC 协议退避时间呈指数递增。除了旁听信道发送之外，攻击者需要的能量极少。

采用差错纠错机制能够容忍消息在任意协议层次上遇到不同程度的损伤。差错纠错编码本身存在额外的处理开销和通信开销。对于一个给定的差错纠错编码，恶意节点仍然能够使其损坏的分组多于网络能够纠正的分组，但是开销较高。

网络可以采用碰撞检测技术来识别恶意碰撞，恶意碰撞会产生一种链路层人为干扰，但是迄今为止还没有彻底有效的防护措施和技术。正当发送仍然需要节点之间的相互协作，以期避免互相损坏对方发送的分组。一个被攻击者彻底颠覆的节点能够故意、反复拒绝信道访问，而其能耗比全时段人工干扰低得多。

2. 能量消耗

链路层可能采用反复重传技术。即使被一个异常延迟的碰撞（如在本帧即将结束时引起的碰撞）所触发的时候，也可能会进行重传。这种主动 DoS 攻击会耗尽附近节点的电池储能，危害网络的可用性（即使攻击者不再进行攻击）。随机退避只能降低无意碰撞概率，却不能防止这种攻击。

时分复接给每个节点分配一个发送时隙，不需要为发送每个帧而进行信道访问仲裁。这种方法能够解决退避算法中的不确定性延迟问题，但是仍然易受碰撞攻击。

可以利用大多数 MAC 协议的交互式特性进行询问攻击。例如,基于 IEEE 802.11 的 MAC 协议采用 RTS/CTS/DATA/ACK 交互方式预留信道访问和发送数据,因此节点可以反复利用 RTS 请求信道访问,得到目标相邻节点的 CTS 响应。持续发送最终耗尽发送节点和目标相邻节点的能量资源。

一种解决方法是限制 MAC 准入控制速率,网络不予理睬过多信道访问请求,不进行能耗甚高的无线发送。这种限制策略不会使准入速率下降到网络所能支持的最大数据速率以下(但是会发生这种情况)。防止电池能量消耗攻击的一个策略是限制无关紧要的、却是 MAC 协议所需要的响应。为了提高总体效率,设计人员常常在系统中实现这种能力,但是处理攻击的软件代码需要额外逻辑。

3. 不公平性

不公平性是一种较弱形式的 DoS 攻击。断断续续地运用碰撞攻击和电池能量消耗攻击,或者滥用协作性 MAC 层优先权机制会引起不公平性。这种安全威胁尽管不能完全阻止合法的信道访问,但是会降低服务质量,如导致实时 MAC 协议的用户发生时间错位。

一种对付不公平性攻击的方法是采用短帧结构,因此每个节点占用信道的时间较短。但是,假如网络经常发送长消息,那么这种方法导致成帧开销上升。在竞争信道访问时,攻击者采取欺骗手段很容易突破这种防护措施:攻击者迅速作出响应,而其他节点则随机延迟其响应。

11.2.3 对WSN网络层（路由）的攻击

由于 WSN 常常依靠电池供电,而电池能量非常有限,所以许多传感器网络路由协议设计得很简单,节省能量,使节点寿命、网络寿命达到最大,因此有时易受攻击。各种 WSN 网络层攻击的主要差异表现在是试图直接操作用户数据的攻击还是试图影响低层路由拓扑的攻击。针对 WSN 进行的网络层攻击分成以下几类:对路由信息的哄骗、篡改、重放;选择性转发;污水池攻击;女巫攻击;蠕虫攻击;hello 泛洪攻击;确认哄骗。下面分别加以分析。

1. 对路由信息的哄骗、篡改、重放

针对路由协议最直接的攻击就是以节点之间交换的路由信息为目标进行攻击。攻击者通过对路由信息的哄骗、篡改、重放,能够创建路由闭环、吸引或者抵制网络流量、延长或者缩短源路由、产生虚假错误消息、分割网络、增大端到端时延等。

2. 选择性转发

多跳网络常常假定参与节点安全、正确地转发所收消息。在选择性转发攻击中,攻击者可能拒绝转发某些消息,简单地将这些消息丢掉,确保这些消息不会进一步传播。当恶意节点的表现类似黑洞、拒绝转发通过其传递的每个分组时,就是这种简单形式的选择性转发攻击。攻击者采用这种形式攻击存在风险:由于接收不到攻击者节点发送的消息,所以相邻节点将会认为攻击者节点已经失效,因而决定寻找另一条路由。另外一种表现形式稍有不同的

选择性转发攻击是：攻击者选择性地转发分组，其兴趣在于抑制或者篡改若干个精选节点产生的分组，但是仍然可靠转发其余流量分组，从而降低了其攻击行为被怀疑的可能性。

当攻击者直接处在数据流传输路由上时，选择性转发攻击通常是非常有效的。攻击者旁听通过相邻节点的数据流量，因此通过人为干扰或者碰撞其感兴趣的每个转发分组就能够模仿选择性转发。这种攻击机制需要高超技巧，因此很难施行这种攻击。例如，如果网络中每个相邻节点对使用唯一一个密钥初始化跳频通信或者扩频通信，那么攻击者要施行这种攻击极其困难。因此，攻击者很可能沿着抗攻击能力最弱的路径，并且尽量包含自身的数据流实际传输路径进行选择性地转发攻击。

3. 污水池攻击

在污水池攻击中，攻击者的目的是引诱来自某个特定区域的附近所有流量通过一个失密节点，从而产生一个比喻性的污水池，中心位置就是攻击者。由于分组传输路径上的节点及其附近的节点有很多机会篡改应用数据，所以污水池攻击能够同时伴随许多其他攻击（如选择性转发攻击）。

污水池攻击的工作原理是使失密节点对路由算法和周围节点看上去很有吸引力。例如，攻击者可以哄骗或者重放到达中心节点的极高质量路由广播消息。有些路由协议可能会采用端到端应答（包含可靠性、时延信息）真正验证路由的质量。此时，微型计算机类攻击者采用大功率发射机直接对中心节点发送（发射功率足够高，单跳可达）或者采用蠕虫攻击，就能够提供到达中心节点的真正高质量路由。由于存在通过失密节点的真正或者虚假高质量路由，所以攻击者的每个相邻节点很可能将传递给中心节点的分组转发给攻击者，并且又将这种高质量路由信息传播给自己的相邻节点。攻击者由此有效创建一个巨大的“影响球”，吸引传递给中心节点的所有数据流（来自离失密节点数个转发跳远的节点）。

进行污水池攻击的一个动机是为了进行选择性地转发攻击。攻击者通过确保特定目标区域的所有数据流传递通过失密节点，就能够选择性抑制或者篡改来自该区域任意节点的分组。

传感器网络特别易受污水池攻击的原因在于其特殊的通信模式。因为所有分组的最终目的节点只有一个中心节点（在只有一个中心节点的 WSN 中），所以失密节点只需要提供单跳可达中心节点的高质量路由就有可能影响大量传感器节点。

4. 女巫攻击

女巫攻击是指一个恶意装置非法占用多个网络身份。将一个恶意装置的额外身份称为女巫节点。女巫攻击会大幅度地降低路由协议、拓扑维护中的容错功效。认为使用不相交节点的各项路由实际上包含冒充多个身份的那个攻击者节点。

一个女巫节点可以采取以下方法获取身份：一种方法是伪造一个新的身份。在有些情况下，攻击者可以简单任意地产生新的女巫身份，例如，假如使用一个 32 bit 的整数表示每个节点的身份，那么攻击者可以给每个女巫节点分配一个随机 32 bit 的整数。另外一种获取身份的方法是窃取某个合法节点的身份。给定一个合法节点身份识别机制，那么攻击者可能无法伪造新的身份。此时攻击者需要将其他合法节点的身份分配给女巫节点。假如攻击者摧毁了假扮节点或者使假扮节点临时性失效，那么可能无法察觉这种身份窃取行为。

女巫节点直接与合法节点通信。当一个合法节点给一个女巫节点发送一条消息时，其

中一个恶意装置在无线信道上侦听此消息。女巫节点发送的消息实际上是其中一个恶意节点发送的。假如合法节点不能与女巫节点直接通信，那么其中一个或者多个恶意装置声明能够到达女巫节点。女巫节点发送的消息通过其中一个恶意节点传递，后者假装将消息传递给女巫节点。

女巫攻击对地理路由协议威胁极大。位置意识路由为了高效地利用地理路由传递分组，一般要求节点与其相邻节点交换位置坐标信息。攻击者运用女巫攻击就能够“立即出现在多个地点”。

5. 蠕虫攻击

一条蠕虫就是一条连接两个网络子区域的低时延链路，攻击者在这条链路上中继网络消息。蠕虫可以由单个节点创建，即该节点位于两个相邻或者不相邻节点之间，转发其间的消息；也可以由一对节点创建，即这两个节点分别位于两个不同的网络子区域，并且相互进行通信。

在蠕虫攻击中，攻击者接收到某个网络子区域的消息，然后沿着低时延链路（蠕虫）将这些消息重放到网络其他区域中。特别是在同一个通信节点对之间，通过蠕虫发送的分组传输时延小于采用正常多跳路由时的分组传输时延。最简单的蠕虫攻击就是一个节点位于另外两个节点之间，转发这两个节点之间的消息。但是，蠕虫攻击通常涉及两个相距较远的恶意节点，这两个恶意节点共同有意低估相互之间的距离，沿着只有攻击者才能够使用的带外信道中继分组。

假如攻击者离中心节点较近，那么攻击者通过精心设计和布置的蠕虫就有可能彻底破坏路由。攻击者可能使离中心节点数个转发跳远的节点相信通过蠕虫只有一跳或者两跳远。这就能够产生污水池：处在蠕虫另一边的攻击者能够提供到达中心节点的虚假高质量路由，要是备用路由没有竞争力，那么附近区域中的所有流量有可能通过蠕虫传递，当蠕虫的端点离中心节点相对较远时就很可能总是如此。

较一般的情况是，蠕虫可以充分利用路由竞争条件。当一个节点根据其接收的第一条消息而忽略随后消息采用某种操作时通常就会出现路由竞争条件。在这种情况下，要是攻击者能够使节点在多跳路由正常到达时间前接收某种路由信息，那么攻击者就能够影响最后得到的拓扑。蠕虫正是这样实现的，即使路由信息被加密和需要认证，蠕虫也仍然有效。蠕虫通过中继两个相距甚远节点之间的分组使这两个节点相信是相邻节点。

蠕虫攻击很可能与选择性转发或者偷听一起使用。当蠕虫攻击与女巫攻击一起使用时，可能很难检测蠕虫攻击。

6. hello泛洪攻击

hello 泛洪攻击就是攻击者利用 WSN 路由协议中使用的 hello 消息进行的攻击。很多 WSN 路由协议要求节点广播 hello 消息，以向其相邻节点声明自己的存在和广播自己的一些信息（如身份、地理位置）。接收到 hello 消息的节点则可假定自己处在该 hello 消息发送节点的覆盖范围内。这个假设条件有可能是虚假的，如微型计算机类的攻击者采用足够大发射功率广播路由或者其他信息，就能够使网络中每个节点相信攻击者就是其相邻节点。

攻击者给每个网络节点广播到达中心节点的质量极高的路由，这样就可能使大量节点使用这条路由，但是离攻击者甚远的所有那些节点发送的分组就会被湮没，从而导致网络处于

混乱状态。节点认识到到达攻击者的这条链路是虚假链路后几乎没有什么可选择的处理办法：其所有相邻节点都可能将分组转发给攻击者。那些依靠相邻节点间位置信息交换来维护网络拓扑或者进行流量控制的协议也易受 hello 泛洪攻击。

攻击者进行 hello 泛洪攻击时不必建立合法分组流。攻击者只需采用足够大的发射功率重复广播开销分组，使每个网络节点能够接收到这个广播。也可以认为 hello 泛洪是单方广播蠕虫。

“泛洪”经常用来表示一条消息在多跳拓扑上迅速传播给每个网络节点。但是 hello 泛洪攻击采用单跳广播将一条消息发送给大量接收节点，所以两者之间是有差别的。

7. 确认哄骗

有些 WSN 路由协议依靠间接或者直接的链路层应答。由于 WSN 传输媒介的固有广播特性，所以攻击者可以旁听传递给相邻节点的分组，并对其做出链路层哄骗应答。应答哄骗的目的包括使发送节点相信一条质量差的链路是一条质量高的链路、一个失效节点或者被毁节点是一个活动节点。例如，路由协议可以运用链路可靠性选择传输路径的下一个转发跳。在应答哄骗攻击中，攻击者故意强迫使用一条质量差链路或者一条失效的链路。因为沿着质量差或者失效链路传递的分组将会丢失，所以攻击者运用应答哄骗能够有效地进行选择性转发攻击，鼓励目标节点在质量差或者失效链路上发送分组。

11.2.4 对传输层的攻击

传输层负责管理端到端连接。传输层提供的连接管理服务可以是简单的区域到区域的不可靠任意组播传输，也可以是复杂、高开销的可靠按序多目标字节流。WSN 一般采用简单协议，使应答和重传的通信开销最低。WSN 传输层可能存在两种攻击：泛洪和去同步。

(1) 泛洪

要求在连接端点维护状态的传输协议易受泛洪攻击，泛洪攻击会引起传感器节点存储容量被耗尽的问题。攻击者不断反复提出新的连接请求，直到每个连接所需的资源被耗尽或者达到连接最大限制条件为止。此后，合法节点的连接请求被忽略。假如攻击者没有无穷资源，那么这是不可能的：攻击者建立新连接的速度快到足以在服务节点上产生资源饥饿问题。

(2) 去同步

去同步就是指打断一个既存连接。例如，攻击者反复给一个端主机发送哄骗消息，使这个主机申请重传丢失分组。假如时间同步正确，那么攻击者可以削弱端主机数据交换能力，甚至阻止端主机交换数据，从而导致端主机浪费能量试图从实际上并不存在的错误中恢复过来。一种对抗措施是要求认证端主机之间通信的所有分组。假定认证方法本身是安全的，那么攻击者就不能给端主机发送哄骗消息。

11.3 SPINS安全解决方案

SPINS 采用安全网络加密协议（Secure Network Encryption Protocols, SNEP）提供数据机密性、数据认证、数据完整性、数据新鲜度，采用 μ TESLA 提供广播认证。

11.3.1 符号

使用如下符号描述 SPIN 及其加密操作：

- A、B 表示主体（比如通信节点）；
 - N_A 表示 A 产生的一个随机数，一个随机数就是一个不可预测的比特串，通常用于实现新鲜度；
 - X_{AB} 表示 A、B 共享的主秘密密钥（对称秘密密钥），主秘密密钥表示符号没有方向性，即 $X_{AB}=X_{BA}$ ；
 - K_{AB} 和 K_{BA} 表示 A、B 共享的秘密加密密钥，A 和 B 根据通信方向和利用主秘密密钥推导秘密加密密钥，即 $K_{AB}=F_{X_{AB}}(1)$ ， $K_{BA}=F_{X_{AB}}(3)$ ，其中 F 表示伪随机函数（Pseudo Random Function，PRF）；
 - K'_{AB} 和 K'_{BA} 表示 A、B 共享的秘密消息认证码密钥。A 和 B 根据通信方向和利用主秘密密钥推导秘密消息认证码密钥： $K'_{AB}=F_{X_{AB}}(2)$ ， $K'_{BA}=F_{X_{AB}}(4)$ ，其中 F 表示伪随机函数；
 - $\{M\}_{K_{AB}}$ 表示采用秘密加密密钥 K_{AB} 加密的消息 M ；
 - $\{M\}_{<K_{AB}, I_V>}$ 表示采用密钥 K_{AB} 和初始矢量 I_V 加密的消息 M ， I_V 用于加密方式[比如加密分组链（Cipher Block Chaining，CBC）、输出反馈方式（Output Feedback Mode，OFM）或者计算方式；
 - $MAC(K'_{AB}, M)$ 表示采用密钥 K'_{AB} 计算消息 M 的消息认证码；
- 一个安全信道就是一个提供机密性、数据认证、完整性、新鲜度的信息。

11.3.2 SNEP

SNEP 具有许多独特优点：SNEP 通信开销低，每条消息只增加 8 B；像许多加密协议一样，SNEP 采用一个计数器，通过在两个端点维持状态而不需要发送计数器值；SNEP 实现语义安全，这是一个很强的安全特性，防止偷听者从加密消息中推导出消息内容；同样简单而高效的协议也能提供数据认证、重放保护、弱消息新鲜度。

数据机密性是最基本安全语义之一，几乎每个安全协议都包含数据机密性。通过加密能够实现简单的机密性，但是纯加密是不够的。另外一个重要安全特性是语义安全，语义安全确保即使偷听者了解同一个明文的若干个密码编码，偷听者仍然不能得到这个明文的任何信息。例如，即使攻击者获得一个加密的 0 bit 和一个加密的 1 bit，但是攻击者仍然不知道一个新加密是 0 还是 1 的加密。安全语义的基本实现技术就是随机化，即在使用链式加密函数（如 DES-CBC）加密消息前，在消息前面插入一个随机比特串。这样能够防止攻击者在知道采用相同密钥的明文-密文对的条件推导出加密消息的明文。

但是，在无线信道上发送随机数据需要较多能量，所以 SPIN 采用另外一种没有额外传输开销的加密机制来实现安全语义：采用两个端点共享的两个计数器（每个通信方向一个计数器）作为 CTR 的分组密码。管理计数器的传统方法是将计数器值与每条消息一起发送。但是由于采用传感器，通行双方共享计数器，每当通信一个分组后就递增计数器，所以发送消

息时不用发送计数器值，从而能够节省能量。

为了实现通信双方的认证和数据完整性，采用消息认证码。对于不同的加密原语不能重复使用相同的加密密钥，防止原语之间的交互，原语交互有可能引入安全弱点。因此，SPIN 推导加密操作和消息认证码操作的独立密钥。两个通信节点 A 和 B 共享一个主秘密密钥 X_{AB} ，并且使用随机函数 F 推导每个通信方向的独立加密密钥和消息认证码密钥：加密密钥 $K_{AB}=F_X(1)$ ， $K_{BA}=F_X(3)$ ， $K'_{AB}=F_X(2)$ ， $K'_{BA}=F_X(4)$ 。

SNEP 就由这些机制有机组成。加密数据格式： $E=\{D\}_{<K,C>}$ ， D 表示数据， K 表示加密密钥， C 表示计数器。消息认证码 $M=MAC(K',C||E)$ 。A 发送给 B 的完整消息为

$$A \rightarrow B: \{D\}_{<K_{AB},C_A>}, MAC(K'_{AB},C_A || \{D\}_{<K_{AB},C_A>}) \quad (11-1)$$

SNEP 具有如下特点：语义安全：因为每当通信完一条消息后递增计数器值，所以同一条消息每次加密均不同，计数器值足够长，在节点存活期间不会重复；数据认证：假如一条消息通过消息认证码验证，那么接收节点知道本条消息是哪个合法发送节点发送来的；重放保护：利用消息认证码中的计数器值能够预防重放过时消息，假如在消息认证码中没有使用计数器，则对手很容易重放消息；弱新鲜度：假如一条消息通过认证，那么接收节点知道本条消息必定是在其前一条正确接收消息（其计数器值较小）之后发送的，从而强化了消息排序，得到弱新鲜度；通信开销低：每个通信端点维护计数器状态，不必随着每条消息发送计数器状态，假如未通过消息认证码验证，那么接收节点可以按照固定小整数递增计数器值，以便从消息丢失中恢复过来；如果仍然未能通过消息认证码验证，那么通信双方执行计数器交换协议。

非加密 SNEP 只强化节点 B 范围内的消息发送顺序，因而只提供弱数据新鲜度，不能对 A 绝对保证节点 B 是响应节点 A 中一个事件而产生的消息。

节点 A 通过 N_A （一个随机数，要求其长度满足不可能穷尽搜索所有可能的随机数）来实现节点 B 的响应的强数据新鲜度。节点 A 随机产生 N_A ，并将其与请求消息 R_A 一同发送给节点 B。实现强新鲜度的最简单方法就是节点 B 按照认证协议将其随机数与响应消息 R_B 一同返回给节点 A。但是，在计算消息认证码的过程中间接使用随机数，优化强新鲜度的实现过程，不需要返回随机数。提供 B 的响应的强新鲜度的完整 SNEP 协议为

$$\begin{aligned} A \rightarrow B: N_A, R_A \\ B \rightarrow A: \{R_B\}_{<K_{BA},C_B>}, MAC(K'_{BA},N_A || C_B || \{R_B\}_{<K_{BA},C_B>}) \end{aligned} \quad (11-2)$$

假如通过消息认证码验证，那么节点 A 知道节点 B 的响应是在 A 发送完请求之后产生的。假如需要机密性和数据认证，那么第一条消息也可以使用非加密 SNEP，如式 (11-1) 所示。

为了实现短小 SNEP 消息，假定通信节点对 A 和 B 相互知道对方的计数器值 C_A 和 C_B ，因此不需要在每条加密消息中增加计数器值。但是在实际中，消息可能丢失，共享计数器状态可能变得不一致。因此，采用计数器交换协议实现计数器状态的同步。为了引导计数器初始值，使用如下计数器值交换协议

$$\begin{aligned} A \rightarrow B: C_A \\ B \rightarrow A: C_B, MAC(K'_{BA},C_A || C_B) \\ A \rightarrow B: MAC(K'_{AB},C_A || C_B) \end{aligned}$$

计数器值不是秘密数据，因此不需要对计数器值加密。但是，计数器交换协议需要强新鲜度，所以通信双方使用其计数器作为随机数（假定计数器交换协议决不会对同一个计数器

值运行两次, 因此必要时可以递增计数器)。消息认证码密钥 K'_{AB} 和 K'_{BA} 间接将本消息约束为通信节点 A 和 B, 并且确保本消息的通信方向, 因此消息认证码不必包含 A 或者 B 的名称。

节点 A 若是知道节点 B 的计数器 C_B 不再是同步的, 则可以使用 N_A 请求 B 的当前计数器, 确保应答的强新鲜度, 即

$A \rightarrow B: N_A$

$B \rightarrow A: C_B, \text{MAC}(K'_{BA}, N_A | C_B)$

为了预防拒绝服务攻击 (攻击者连续发送虚假消息, 欺骗节点执行计数器同步), 节点可以利用其发送的每条加密消息来发送计数器值。另外一种计数器状态同步 DoS 攻击的检测方法是在消息中增加一个新的、与计数器无关的较短消息认证码。

11.3.3 μ TESLA

TESLA 协议提供高效广播认证。但是, TESLA 不是为传感器网络这种有限计算环境设计的, 理由如下:

- TESLA 采用数字签名认证初始分组, 对于传感器节点, 数字签名计算开销太高, 因为即使将数字签名程序代码安装到存储器中也是面临的一个挑战;
- 标准 TESLA 每个分组约 24 B 开销, 对于连接工作站的网络, 这个开销是无关紧要的, 但是传感器节点发送极短消息, 大约每条消息 30 B, 暴露每个分组前一个间隔的 TESLA 密钥完全不可能, 由于 64 bit 密钥以及消息认证码, 所以 TESLA 有关部分占一个分组长度的 50% 以上;
- 单向密钥系列不适合传感器节点的存储器, 所以 TESLA 不适用于无线传感器节点广播数据认证。

μ TESLA 用于解决 TESLA 用于 WSN 中面临的如下问题和不足:

- TESLA 采用数字签名认证初始分组, 这种方法对于 WSN 节点代价太高, μ TESLA 只采用对称机制;
- 暴露每个分组的密钥需要太多发送能量和接收能量, μ TESLA 按照每个时段暴露一次密钥;
- 在 WSN 节点中存储单向密钥系列代价高, μ TESLA 限制待认证节点的数量。

广播认证需要非对称机制, 否则任何有安全风险的节点就能够伪造发送节点发送的消息。但是, 非对称加密机制存在很高的计算、通信、存储开销, 因此不能用于资源很有限的 WSN 装置。 μ TESLA 可以克服这个问题: 延迟对称密钥的暴露时间, 由此引入非对称性, 得到一个高效广播认证方案。

μ TESLA 要求中心节点和 WSN 节点之间松散时间同步, 每个节点知道最大同步误差的上限。为了发送一个待认证分组, 中心节点利用一个当时是秘密的密钥计算该分组消息认证码。一个节点接收到一个分组后, 根据其松散同步时钟、最大同步误差以及密钥暴露的时间表就能够验证该分组的消息认证码密钥还未被暴露。由于对接收节点确保只有中心节点知道该消息认证码密钥, 所以对接收节点保证该分组在传输途中不会被对手所篡改。节点将该分组存储在缓存器中。到达密钥暴露时刻, 中心节点给所有接收节点广播验证密钥。一个节点接收到暴露密钥后, 就能够验证密钥的正确性。假如密钥验证正确, 那么节点就可以使用该密钥认证其缓存器中存储的那个分组。

每个消息认证码密钥就是一个密钥链中的一个密钥，是通过公共单向函数 F 产生的。为了产生单向密钥链，发送节点随机选择该序列中最后一个密钥 K_n ，对其反复进行 F 运算，计算出所有其他密钥： $K_i=F(K_{i+1})$ 。每个节点按照安全、认证方式，运用 SNEP 安全模块，易于实现时间同步和重新得到单向密钥链中待认证密钥。

例如，图 11-1 表示 μ TESLA 单向密钥链导出、时间间隔、发送节点广播的若干分组。单向密钥链中的每个密钥对应一个时间间隔，在一个时间间隔内发送的所有分组采用同一个密钥进行认证。发送节点每隔两个时间间隔就暴露其用来计算消息认证码的密钥。假定接收节点松散时间同步，并且知道 K_0 （该单向密钥链的第一个密钥）。在时间间隔 1 发送的分组 P_1 和 P_2 包含一个消息认证码和密钥 K_1 。在时间间隔 2 发送的分组 P_3 包含一个消息认证码和密钥 K_2 。至此，接收节点还不能认证任何分组。假定分组 P_4 、 P_5 、 P_6 全部丢失，暴露密钥 K_1 的分组也丢失，因此接收节点仍然不能认证分组 P_1 、 P_2 或者 P_3 。中心节点在时间间隔 4 广播密钥 K_2 ，节点通过验证 $K_0=F[F(K_2)]$ 来认证密钥 K_2 。节点推导 $K_1=F(K_2)$ ，所以能够使用密钥 K_1 认证分组 P_1 、 P_2 ，使用密钥 K_2 认证分组 P_3 。

密钥暴露与分组广播无关，但是受时间间隔约束。在 μ TESLA 中，发送节点在一个特殊分组中周期性广播当前密钥。

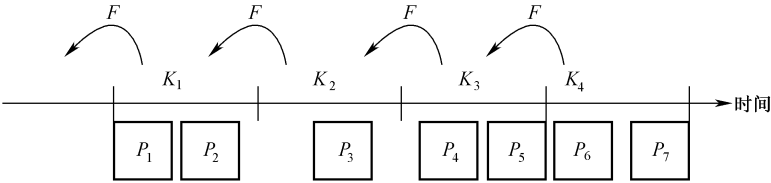


图 11-1 μ TESLA 单向密钥链

11.3.4 μ TESLA详细描述

μ TESLA 包含多个阶段：发送节点建立阶段、待认证分组发送阶段、新接收节点引导阶段、分组认证阶段。

1. 发送节点建立阶段

发送节点首先产生一个秘密密钥链（单向密钥链）。为了产生长度为 n 的单向密钥链，发送节点随机选择最后一个密钥 K_n ，连续进行单向函数 F 运算，产生剩余密钥 $K_j=F(K_{j+1})$ 。由于 F 是单向函数，所以任何人都能够做前向计算，比如已知 K_{j+1} 就能够计算出 K_0 、 K_1 、 \cdots 、 K_j ；但是由于生成函数是单向的，所以任何人不能进行后向计算，比如已知 K_0 、 K_1 、 \cdots 、 K_j ，不能计算出 K_{j+1} 。

2. 待认证分组广播阶段

时间被划分成均匀的时间间隔，发送节点使单向密钥链中的每个密钥关联一个时间间隔。在时间间隔 i 内，发送节点采用该间隔的密钥 K_i 计算该间隔上各个分组的消息认证码。在时间间隔 $(i+\delta)$ 内，发送节点暴露密钥 K_i 。要求密钥暴露时延大于发送节点与任意接收节点之间的往返传输时延，因此密钥暴露时延是若干个时间间隔的 10 倍。

3. 新接收节点引导阶段

在单向密钥链中, 密钥是自行认证的。接收节点使用一个已认证的密钥, 很容易且又高效地认证其余的密钥。例如, 接收节点接收到单向密钥链中的密钥 K_i , 则容易认证 K_{i+1} , 验证 $K_i = F(K_{i+1})$ 。为了引导 μ TESLA, 每个接收节点必须将其中一个密钥作为整个单向密钥链的第一个密钥。其他要求是发送节点和接收节点松散时间同步, 接收节点知道单向密钥链中各个密钥的暴露时间表。采用强新鲜度机制和点对点认证能够建立松散时间同步和待认证密钥链的第一个密钥。接收节点 R 通过请求消息给发送节点 S 发送 N_R 。发送节点 S 应答的消息包含其当前时间 T_S 、在前一个时间间隔 i 使用的密钥 K_i (该单向密钥链的第一个密钥)、时间间隔 i 的起始时间 T_i 、一个时间间隔的长度 T_{int} 、暴露时延 δ (T_i 、 T_{int} 、 δ 是暴露时间表的描述), 即

$$\begin{aligned} M \rightarrow S: & N_M \\ S \rightarrow M: & T_S | K_i | T_i | T_{int} | \delta \\ & MAC(K_{MS}, N_M | T_S | K_i | T_i | T_{int} | \delta) \end{aligned}$$

因为不要求机密性, 所以发送节点不必加密该数据。消息认证码使用该节点与中心节点共享的秘密密钥认证该数据, N_M 允许该节点验证新鲜度。没有采用 TESLA 的数字签名技术, 而是采用节点到中心节点的待认证信道引导待认证广播。

4. 广播分组认证阶段

接收节点接收到包含该消息认证码的分组后, 必须确保该分组不是对手的哄骗分组。对手已经知道一个时间间隔的暴露密钥, 并且知道使用这个密钥来计算这个消息认证码, 所以能够伪造这个分组。接收节点必须确保这个分组是安全的, 这意味着发送节点还没有暴露用于计算输入分组的这个消息认证码的这个密钥。发送节点和接收节点必须松散时间同步, 接收节点必须知道密钥暴露时间表。假如输入分组是安全的, 那么接收节点存储这个分组 (一旦其对应密钥被暴露, 就能够验证该分组)。假如输入分组不是安全的 (该分组经历异常长的时延), 那么该分组很可能已经被对手篡改过, 因此接收节点丢掉该分组。

接收节点一旦接收到一个新密钥 K_i , 则认证该密钥: 进行少数几次单向函数 F 运算 $K_v = F^{i-v}(K_i)$, 检查 K_i 是否与其最近所知的可信密钥 K_v 匹配。假如两个密钥匹配, 那么新密钥 K_i 是可信密钥, 接收节点可以认证在时间间隔 v 至 i 上发送的所有分组。接收节点还要用 K_i 替换已存储的 K_v 。

5. 节点广播通过认真的分组

假如一个节点广播一个已通过认证的数据, 则会产生新的挑战, 其原因是: 节点存储器容量有限, 不能存储单向密钥链的各个密钥; 根据初始生成的密钥 K_n 重复计算每个密钥的计算开销很高; 一个节点可能不会与其所有相邻节点共享同一个密钥, 因此发送该密钥链已通过认证的第一个密钥涉及代价甚高的点对点密钥协议; 节点将已暴露密钥广播给所有接收节点代价甚高, 耗尽其已有电池能量。有两种方法解决这个问题: ①节点通过中心节点来广播其数据, 节点按照认证方式采用 SNEP 将其数据发送给中心节点, 中心节点接收到该数据后再广播该数据; ②节点广播其数据, 中心节点保存单向密钥链, 按需给广播节点发送密钥。为了节省广播

节点的能量，中心节点也广播暴露密钥，同时（或者）执行新接收节点的初始引导规程。

11.3.5 SPINS实现

采用 SmartDust 传感器平台（其配置如表 11-1 所示）来实现 SPINS。其中 8 KB 只读程序存储器用于存储 TinyOS、SPINS 以及传感器网络应用的程序代码。

表 11-1 SmartDust 传感器节点的特征

CPU	8 bit, 4 MHz
存储器	程序存储器: 8 KB
	RAM: 512 B
	EEROM: 512 B
通信	916 MHz 电台
带宽	10 kb/s
操作系统	TinyOS
操作系统代码长度	3 500 B
用户可用的代码存储器容量	4 500 B

(1) 分组密码

采用 RC5^[2]。RC5 代码小、效率高，没有采用乘法，不需要大表格。但是，RC5 采用 32 bit 宽的数据循环操作（而采用的 SmartDust 传感器平台只支持 8 bit 单比特循环操作）。尽管能够成功表达 RC5 算法，但是 RC5 公共库太大而不能安装到 SmartDust 传感器平台上，因此从 RC5 公共库中选择一部分，将代码额外减小 40%。

(2) 加密函数

为了节省代码存储空间，加密、解密采用相同函数。分组密码的计数器（CounTeR, CTR）方式具有这种特性，如图 11-2（a）所示。CTR 方式是流密码。因此，密文的长度正好等于明文的长度，而不等于分组密码长度的整数倍。对于 WSN 环境特别需要这个特性。消息发送和接收消耗很多能量，消息越长，数据被损坏的概率就越高。CTR 方式要求计数器正确操作。重复使用计数器值严重影响安全性。CTR 方式提供语义安全：由于密码填充是根据不同计数器产生的，所以不同时间发送的相同明文采用不同的密码加密。对于不知道密钥的攻击者，这些消息似乎是两个不相关的随机串。由于发送节点和接收节点共享一个计数器，所以消息中不必包含计数器。假如两个节点的计数器不同步，那么这两个节点可以运用 SNEP 和强新鲜度直接发送计数器，重新实现同步。

(3) 新鲜度

CTR 加密自动提供弱新鲜度。由于发送节点在每条消息之后递增计数器，所以接收节点通过检验所收消息具有单调递增计数器就可验证弱新鲜度。对于要求强新鲜度的应用，发送节点随机生成一个 N_M （一个不可预测的 64 bit 数值），并将这个 N_M 添加到发送给接收节点的请求消息中。接收节点产生响应消息，并且其消息认证码计算中包含这个 N_M 。假如响应的消息认证码通过验证，那么发送节点知道这个响应是在其发送请求消息之后产生的，从而实现强新鲜度。

(4) 随机数的生成

节点有其自己的传感器、无线接收机以及时间安排过程，因此可以据此推导随机数。为

为了使能量需求最低，采用消息认证码函数作为伪随机数生成器（Pseudo-Random Number Generator, PRG），同时采用秘密伪随机数生成器密钥 X_{rand} 。维持一个计数器 C ，每当生成一个伪随机分组后就将 C 加 1。计算第 C 个伪随机输出分组为 $\text{MAC}(X_{\text{rand}}, C)$ 。假如 C 发生卷绕（实际上不可能发生，因为节点在此之前首先会耗尽其能量），那么可以根据主秘密密钥和当前 PRG 密钥，采用消息认证码函数作为伪随机函数（Pseudo-Random Function, PRF），就可推导出一个新的 PRG 密钥，即 $X_{\text{rand}} = \text{MAC}(X, X_{\text{rand}})$ 。

（5）消息认证

需要安全的消息认证码。因为要求重复使用分组密码，所以采用 CBC-MAC^[1]。CBC-MAC 的计算方框图如图 11-2（b）所示。

采用以下权威方法实现消息认证和消息完整性。假定一条消息 M 、一个加密密钥 K 、一个消息认证码密钥 K' ，那么使用结构： $\{M\}_K, \text{MAC}(K', \{M\}_K)$ 。这个结构防止节点解密错误密文，解密错误密文可能存在安全风险。

对每个分组计算消息认证码。这种实现方法非常适合 WSN 环境中 有损通信，而且消息认证码用于检验消息的认证和完整性，因此不需要诸如 CRC 之类的机制。

（6）密钥设置

SPINS 密钥建立依赖主秘密密钥，开始时由中心节点和一个传感器节点共享主秘密密钥。将节点 A 和中心节点 S 共享的主秘密密钥称为共享密钥 X_{AS} ，其他密钥都是从初始主秘密密钥推导出来的。图 11-2（c）给出了 SPINS 密钥推导规程。采用 PRF 函数推导密钥： $F_K(x) = \text{MAC}(K, x)$ ， F 是 PRF 函数。因此可重复使用的分组密码更多。由于消息认证码的加密特性，所以 F 必须是一个良好的伪随机函数。按照这种方法推导出来的所有密钥都是独立计算出来的。即使攻击者能够攻克其中一个密钥，但是利用这个密钥的信息也无法帮助攻击者找到主秘密密钥和其他密钥。假如检测出一个密钥已经不安全，那么通信双方可以重新推导出一个新密钥，而无需发送任何秘密信息。

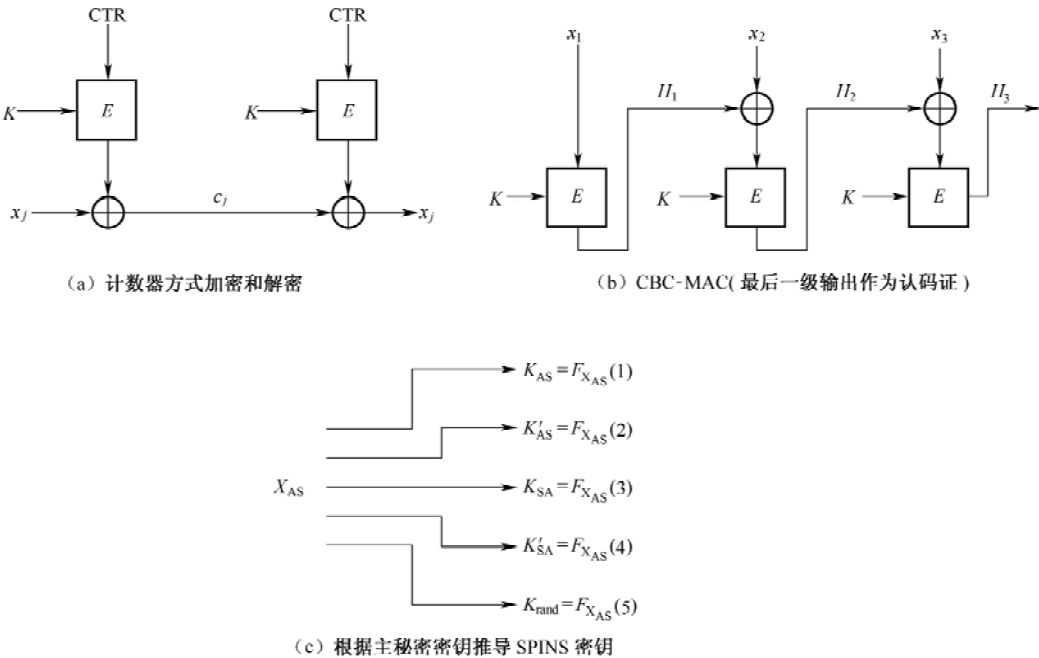


图 11-2 SPINS 实现

11.3.6 SPINS性能评估

1. 密码大小

表 11-2 给出了在 TinyOS 中安全程序三种实现的代码长度。最小版实现约占 SmartDust 传感器平台程序存储器容量的 20%。最小版实现和最快版实现之间的差异在于变量循环函数的实现不同。 μ TESLA 协议代码额外占用 574 B。加密库和 SPINS 协议总共约占 2 KB 的程序存储器容量，这对于大多数 WSN 应用是可以接受的。

表 11-2 安全模块的代码大小清单（单位：字节）

版本	总长度	消息认证码	加密	密钥建立
最小版	1 580	580	402	598
最快版	1 844	728	518	598
原始版	2 674	1 210	802	686

2. 性能

表 11-3 给出了安全的时间性能。从表 11-2 中可以看到：密钥建立的时耗相对较大（约 4 ms）；最快版完成加密一条 16 B 消息和计算其消息认证码总时耗不足 2.5 ms，而最小版时耗不足 3.5 ms。因为 SmartDust 传感器节点的通信速率为 10 kb/s，所以对于发送的每条消息可以执行完密钥建立、加密以及一个消息认证码计算。

各个安全模块的 RAM 占用量分别是：RC5 占用 80 B， μ TESLA 占用 120 B，加密/消息认证码占用 20 B。

表 11-3 安全的时间性能

操 作	所需时间/ms	
	最快版	最小版
加密（16 B）	1.10	1.69
消息认证码（16 B）	1.28	1.63
密钥建立	3.92	3.92

3. 能耗

按照 30 B 分组计算能耗。表 11-4 给出了 SNEP 协议的计算与通信的能量开销。发送能耗明显高于计算能耗。由于采用流密码进行加密，所以加密消息的长度等于明文的长度。给每条消息增加 8 B 消息认证码开销。由于消息认证码具有完整性保证，所以不需要占 2 B 的 CRC，因此净开销 6 B，发送这 6 B 的能耗是发送一个 30 B 分组总能耗的 20%。

采用 μ TESLA 进行消息广播的能耗等于每条消息的认证能耗。此外， μ TESLA 要求周期性密钥暴露，但是这些消息与路由更新组合在一起。假如 WSN 必需路由信标，那么 μ TESLA 密钥暴露几乎没有能耗，这是因为发送与接收的能耗明显高于计算能耗。假如路由信标不是 WSN 必需的，并且能够间接建立 Ad Hoc 多跳网络，那么密钥暴露开销是每个时间间隔一条

消息，而与网内流量模式无关。

表 11-4 给 WSN 增加安全协议后的能耗

操 作	能 耗 比
发送数据	71%
发送消息认证码	20%
发送随机数（为了新鲜度）	7%
消息认证码和加密计算	2%

11.4 LEAP+安全解决方案

本地加密与认证协议（Localized Encryption and Authentication Protocol, LEAP）是传感器网络的一个密钥管理协议，具有如下特点：

- LEAP+支持每个节点建立四种密钥：与中心节点共享的单独密钥、与每个相邻节点共享的成对密钥、与一组相邻节点共享的分群密钥、与所有网络节点共享的全网密钥，使用这四种密钥能够提高很多协议的安全性；
- LEAP+包含一个高效弱本地广播认证协议，该协议采用单向密钥链；
- LEAP+的一个显著特征是 LEAP+的密钥共享方法支持网内处理，同时将节点失密引起的安全影响限制在失密节点的直接相邻区域内；
- LEAP+能够防止对传感器网络的许多攻击，还能够加大进行许多攻击的困难；
- LEAP+采用的密钥建立规程和密钥更新规程是高效的，对每个节点的存储要求也很低。

LEAP++提供多种密钥机制，用于实现传感器网络的机密性和认证。实验表明：LEAP+是一个实用的、资源高效的协议，能够保护许多传感器网络应用和协议。

11.4.1 假设条件

假定固定传感器网络，传感器节点固定不动。作为网络控制器（或者密钥服务器）的中心节点是微型计算机类装置，具有持续供应的电能。各个传感器节点具有类似于当代传感器节点（比如伯克利 Mica 传感器）的计算能力、通信能力以及能量资源。每个传感器节点能够保存数百个字节的密钥信息。传感器节点可以采取空中撒播，也可以采取人工安装。事先不知道任何传感器节点的直接相邻节点。

因为无线通信采用广播传输媒介，所以假定攻击者能够偷听所有流量、注入分组或者重放过时消息。假定一个节点失密，那么攻击者知道该节点保存的所有信息。中心节点是安全的，不会失密。LEAP+不是对抗物理层攻击和媒介访问控制层攻击的协议。

11.4.2 LEAP+概述

根据不同分类准则，传感器网络节点之间交换的分组有不同的归类，如控制分组与数据分组，广播分组与单目标分组，查询、命令与传感器感知数据等。一个分组的安全要求通常

与其归类有关。各种类型的分组都需要进行认证，而有些类型的分组才需要机密性。例如，路由控制信息通常不要求机密性，而传感器报告的（累积）感知数据、中心节点发送的查询则需要机密性。

单靠一种密钥机制解决传感器网络中所要求的所有安全通信是不可能的。**LEAP+**支持每个传感器节点建立四种类型的密钥：与中心节点共享的单独密钥、与另一个传感器节点共享的成对密钥、与多个相邻节点共享的分群密钥、与所有网络节点共享的全网密钥。

(1) 单独密钥

每个传感器节点有一个与中心节点共享的唯一的单独密钥。这个密钥用于该节点与中心节点之间的安全通信。例如，假如一个传感器节点的感知数据需要接受中心节点的检验，那么该节点可以采用其单独密钥计算该感知数据的消息认证码。一个传感器节点若是察觉到其某个相邻节点表现异常或者意外，则可以给中心节点发送一个告警。同理，中心节点可以利用与某个传感器节点共享的单独密钥加密敏感信息（如密钥、特殊指令），再将其发送给这个传感器节点。

(2) 成对密钥

每个节点与其每个直接相邻节点共享一个成对密钥。在 **LEAP+**中，成对密钥用于要求秘密或者源认证的安全通信。例如，一个节点可以利用其成对密钥实现其分群密钥安全分发给其相邻节点，或者将其传感器感知数据安全发送给累积节点。利用成对密钥可以排除被动参与。

(3) 分群密钥

一个分群密钥是一个节点与其所有相邻节点共享的密钥，用于本地消息（如路由控制消息）的安全广播或者能够从被动参与中获得的传感器安全消息。**LEAP+**给每个节点提供一个唯一分群密钥；该节点与其所有相邻节点共享此分群密钥，利用此分群密钥加密发送给相邻节点的消息；其相邻节点利用这个分群密钥解密、认证该节点发送来的消息；其每个相邻节点也均有自己的分群密钥，用于加密发送给自己相邻节点的消息。

(4) 全网密钥

这是一个所有网络节点与中心节点共享的秘密密钥，主要用于中心节点加密广播给整个网络的消息。例如，中心节点广播给整个网络的查询或者命令。从机密性来看，利用每个节点的单独密钥单独加密一条广播消息不存在任何优势。但是，由于全网密钥是与所有网络节点共享的密钥，所以当失密节点被排除后。需要密钥重新建立机制来更新全网密钥。

全网密钥的建立（以及重新建立）协议采用分群密钥，而分群密钥则采用成对密钥来建立（以及重新建立）。下面描述 **LEAP+**提供的上述密钥建立与更新机制。在描述过程中，使用了如下符号： N 表示网络中的传感器节点数量； u 、 v 表示通信节点； $\{f_k\}$ 表示一簇伪随机函数； $\{s\}_k$ 表示利用密钥 k 的加密消息 s ； $MAC(k,s)$ 表示利用对称密钥 k 的消息 s 的消息认证码。

节点可以从一个密钥 K 推导出其他密钥，达到各种安全目的。例如，一个节点可以使用 $K'=f_k(0)$ 进行加密、使用 $k'=f_k(1)$ 进行认证。为了简化说明，下面简单地说明使用密钥 K 加密或者认证一条消息，而不管实际上是利用 K' 加密该消息而利用 k'' 认证该消息。

11.4.3 单独密钥的建立

每个传感器节点都有一个只与中心节点共享的单独密钥。单独密钥是在传感器网络布置

之前产生的，并事先加载到每个传感器节点中。

节点 u 的单独密钥 IK_u 生成如下： $IK_u = f_{K^m}(u)$ ， f 是一个伪随机函数， K^m 是一个只有网络控制器知道的主密钥。控制器只保存其主密钥，以便节省保存所有单独密钥所需要的存储容量。当需要与单个传感器节点 u 通信时，控制器即时计算 IK_u 。由于伪随机函数计算效率高，所以本步的计算开销可忽略不计。

11.4.4 成对密钥的建立

在任何传感器网络应用中都需要传感器节点与其直接相邻节点之间的通信。因此，成对密钥是使用最普遍的。下面介绍两种方法建立成对密钥：基本方案和扩充方案。扩充方案的安全性能较强，但是代价是实现复杂性较高、存储需求较大。

1. 基本方案

若能够预先确定传感器节点的相邻节点（如通过人工安装传感器节点时事先通过周密计划和安排），那么就能够预先加载成对密钥。但是，在传感器网络布置完毕之前无法预先知道相邻节点时（如通过空中播撒传感器节点），如何建立成对密钥？下面回答这个问题。

LEAP+利用由固定节点组成的传感器网络的特定属性。第一，一个传感器节点的相邻节点相对固定，增加到网络中的传感器节点在其初始布置时候能够找到其大部分相邻节点。第二，布置在安全关键环境中的一个传感器节点被攻击者捕获后，肯定可以被伪造，以便进行至少可以持续较短时间（如几秒）的非法入网攻击；否则，攻击者可能很容易危及所有传感器节点的安全，然后接管整个网络。因此，不是假定传感器节点具有防篡改能力（这个能力常常得到虚假结果^[3]），而是假定存在一个时间间隔下限值 T_{\min} ， T_{\min} 对于攻击者危及一个传感器节点安全是必需的；还假定一个新布置的传感器节点发现其直接相邻节点所需时间 T_{est} 小于 T_{\min} 。实际实验指出^[5]：在正确实验和合适工具下，一个传感器节点失密后，可能只需要几十秒或者几分钟就能够获取一个 Mica2 传感器中所有存储信息和数据。

LEAP+成对密钥基本建立方法包括四个步骤：

(1) 密钥预分发

控制器生成一个初始密钥 K_{IN} 并将其加载到每个节点中。每个节点 u 推导一个主密钥 $K_u = f_{K_{\text{IN}}}(u)$ 。

(2) 相邻节点寻找

节点 u 布置完毕后，立即广播 hello 消息（包含 u 的 ID），寻找其相邻节点。节点 u 启动一个定时器，定时时间设为 T_{\min} ，定时结束时将会触发密钥删除。节点 u 等待其每个相邻节点 v 的 ACK 响应消息（包含响应相邻节点的 ID，即 v ）。使用相邻节点 v 的主密钥 K_v 认证其 ACK， $K_v = f_{K_{\text{IN}}}(v)$ 。节点 u 知道 K_{IN} ，因而能够推导出 K_v ，然后就能够验证相邻节点 v 的身份。

$$u \rightarrow *: u$$

$$v \rightarrow u: v, \text{MAC}(K_v, u|v)$$

(3) 成对密钥建立

节点 u 计算其与相邻节点 v 的成对密钥 K_{uv} ： $K_{uv} = f_{K_v}(u)$ 。节点 v 也采用相同方法计算其

与 u 的成对密钥 K_{uv} 。 K_{uv} 作为 u 、 v 的成对密钥。在这步操作中 u 、 v 之间没有交换消息。节点 u 不必通过向节点 v 发送特殊消息来认证自己，这是因为节点 u 利用 K_{uv} 进一步发送的任何消息都能证明 u 的身份。

(4) 密钥删除

节点 u 的定时器经过 T_{\min} 后定时结束时，节点 u 删除 K_{IN} 以及其相邻节点的所有主密钥 (K_v)。但是，节点 u 不会删除自己的主密钥 K_u 。每个节点只保存自己的主密钥。

完成以上各步操作后，节点 u 完成建立与其每个相邻节点共享的成对密钥， u 与每个相邻节点可以利用这个成对密钥进行安全的数据交换。在两个方向（即 u 至相邻节点的通信、相邻节点至 u 的通信）上采用相同的成对密钥进行安全通信。网络中所有节点都会删除 K_{IN} 。攻击者在本阶段期间可能偷听所有流量，但是因为缺少 K_{IN} 而无法向网络注入错误信息，也无法解密任何消息。攻击者攻克一个传感器节点后再经过 T_{\min} 后只能获取这个失密节点的密钥信息，而不能获取任何其他节点的密钥信息。当检测到一个失密节点时，其相邻节点删除与该失密节点共享的密钥。

当两个相邻节点 u 和 v 同时被增加到网络中时，可以简化上述成对密钥基本建立方法。例如，假如 u 在响应 v 发送的 hello 消息之前接收到 v 对自己所发 hello 消息的响应，那么 u 将抑制自己的响应。但是，假如 u 和 v 分别单独完成其相邻节点寻找过程，那么在成对密钥建立阶段将会得到两个不同的成对密钥 K_{uv} 和 K_{vu} 。此时，若 $u < v$ ，则选择 K_{uv} 作为 u 、 v 的成对密钥。

性能分析：LEAP+成对密钥基本建立方法的开销分析如下。入网节点必须验证其每个相邻节点的一个消息认证码以及评估生成其成对密钥的伪随机函数。每个相邻节点计算一个消息认证码，生成一个成对密钥。建立一个成对密钥的通信开销包括一条 ACK 消息。ACK 消息有两个组成域：一个节点 ID、一个消息认证码。一条 hello 消息只包含一个节点 ID。ACK、hello 很容易封装到一个分组中。另外只要求提供一个密钥 K_{IN} 的存储空间。因此，LEAP+成对密钥基本建立方法的计算开销、通信开销、存储开销非常低。

安全分析：在 LEAP+成对密钥基本建立方法中做出的主要假设条件是完成相邻节点寻找阶段所需的实际时间 T_{est} 小于 T_{\min} 。这个假设条件对于许多传感器网络和安全攻击是合理的。目前传感器节点的发送速率达到 19.2 kb/s；ACK 消息非常短：假如一个 ID 长 4 B，一个消息认证码长 8 B，那么一条 ACK 消息长 12 B。信道不可靠以及传输碰撞引起的分组丢失是 LEAP+面临的一个挑战。稍后介绍几种技术降低分组丢失率，以便传感器节点在 T_{\min} 内即使不能与其所有相邻节点，也能够与其大多数相邻节点建立起成对密钥。

在上述 LEAP+成对密钥基本建立方法中，对 hello 消息未作认证。攻击者可以利用这一点注入大量 hello 消息，进行资源消耗攻击。一个节点对于从一个新相邻节点接收到的每条 hello 消息，必须计算一个消息认证码，回送一条 ACK 消息，在相邻节点表中维护一个条目。从而消耗了 CPU 时间、通信资源以及存储资源。消息开销比计算开销高得多。参考文献[7]指出：计算一个消息认证码的能耗相当于发送一个字节的能耗。此外，这种资源消耗攻击会引起缓存器溢出，不仅是因为传感器节点 RAM 容量极其有限，而且还因为 TinyOS 只支持固定存储器分配。

采用两种方法来减轻这种资源消耗攻击。这两种方法要求计算每条 hello 消息的消息认证码，但是能够防止这种攻击消耗通信资源和存储资源。第一种方法是，网络控制器可以利用当前全网密钥（初始网络密钥 K_{IN} 除外）预先加入一个新传感器节点。一个新节点可以利

用当前全网密钥认证发送给其相邻节点的 hello 消息。欺诈性 hello 消息将被检测出来而被丢掉，因此不会发送 ACK 消息，也不会维护失密相邻节点的状态。这种方法只能预防外部攻击者的攻击。内部攻击者若是在布置初期就攻克一个传感器节点，就有可能知道当前全网密钥。因此，第二种解决方法就是在新入网节点的 ID 上增加一些随机性，以便检测出欺诈性 ID 并将其丢掉。

为了提高攻击者捕获一个传感器节点后恢复 K_{IN} 的难度，节点只要一加电就立即将 K_{IN} 从永久存储器复制到非永久性存储器，同时删除永久存储器中的 K_{IN} 复制。这就意味着存在以下几个假设条件：一是传感器节点能够彻底删除一个密钥；二是传感器节点 u 不会保存另一个传感器节点 v 的主密钥。只要加载到传感器节点中的程序正确执行，那么这种情况将不会发生。

在 LEAP+中，只有最近布置的传感器节点才拥有 K_{IN} ，能够建立与其相邻节点的密钥。一个节点一旦删除了 K_{IN} ，就不能建立与也已经删除了 K_{IN} 的任何其他节点的成对密钥。这有助于防止节点繁殖攻击和节点复制攻击^[8]。在节点繁殖攻击中，攻击者对自己的节点加载失密节点的密钥，然后将这些克隆节点布置到传感器网络中的不同地点。这些克隆节点试图建立与其相邻节点的成对密钥，一旦被其相邻节点所接受，就能够进行各种内部攻击（如注入虚假数据分组）。结果，由于传感器网络无人照看，所以攻击者可能只需要攻克少数几个传感器节点就能够攻垮整个网络。LEAP+抗节点繁殖攻击能力强的理由是克隆节点没有 K_{IN} ，因此不能建立与非失密节点的相邻节点的节点的成对密钥。因此，LEAP+将失密节点的安全影响限制在其局部范围内。假如节点在布置之前就失去了安全，那么 LEAP+成对密钥基本建立方法不能防止节点繁殖攻击。攻击者能够引入具有相同或者不同节点 ID 的新节点。但是在相同假设条件下，对于其他成对密钥建方法，攻击者比较难以使节点具有不同的 ID。

2. 扩充方案

在 LEAP+成对密钥基本建立方法中，新增加的一个传感器节点利用初始网络密钥 K_{IN} 推导其主密钥和与其相邻节点共享的成对密钥。假如传感器节点在时间 T_{min} 内不会失密，那么因为 K_{IN} 不会失密，所以 LEAP+成对密钥基本建立方法是安全的。现在考虑更加有力的攻击，假定一个传感器节点在时间 T_{min} 内失密或者突破任务权威的安全，那么 K_{IN} 失密。其结果是网络中所有成对密钥失密，而且攻击者知道 K_{IN} ，因而能够将自己的新节点增加到传感器网络中。下面介绍对抗这些攻击的几种措施。

第一种方法是即使攻击者已经设法获得初始密钥 K_{IN} ，也仍然防止攻击者推导主密钥和与布置初期或者布置之后的节点共享的成对密钥。基本思想是排除依赖单个初始密钥 K_{IN} ，而是利用一个初始密钥链来推导各个节点的主密钥。

假定在一个传感器网络应用中最多存在 M 个增加节点事件，这 M 个事件分别发生在 M 个间隔 T_1 、 T_2 、 \cdots 、 T_M 内，这些时间间隔可能长度互不相同。网络控制器随机生成 M 个密钥： K_{IN}^1 、 K_{IN}^2 、 \cdots 、 K_{IN}^M ；这些密钥作为初始密钥。图 11-3 给出了密钥与时间间隔之间的对应关系。下面描述初始化一个新节点 u ，并在时间间隔 T_i 将其增加到传感器网络中。

T_1	T_2	T_3	\cdots	T_{M-1}	T_M
K_{IN}^1	K_{IN}^2	K_{IN}^3	\cdots	K_{IN}^{M-1}	K_{IN}^M

图 11-3 将传感器网络的寿命划分成 M 个时间间隔，每个时间间隔有一个初始密钥

(1) 密钥预分法

在时间间隔 T_i 布置的节点 u 加载有初始密钥 K_{IN}^i ，并利用 K_{IN}^i 推导自己在当前时间间隔 T_i 的主密钥 $K_u^i = f_{K_{IN}^i}(u)$ ；节点 u 还加载有主密钥 $K_u^j = f_{K_{IN}^j}(u)$ ，对于随后所有时间间隔 $i < j \leq M$ ，但是没有加载对应于这些时间间隔的初始密钥 K_{IN}^j 。

(2) 相邻节点寻找

当布置节点 u 时，节点 u 广播一条 hello 消息，初始化一个定时器，定时时间设为 T_{\min} 。hello 消息包含节点 u 的 ID 以及时间间隔 ID（即 i ）。每个相邻节点 v 回送 ACK 消息，ACK 消息包含响应相邻节点 v 的 ID。利用每个相邻节点的当前主密钥 K_v^i 认证其回送的 ACK 消息。节点 u 知道 K_{IN}^i ，因而能够推导出 K_v^i ，然后就能够验证 ACK 消息。

$$\begin{aligned} u \rightarrow *: & u, i \\ v \rightarrow u: & v, \text{MAC}(K_v^i, u|v) \end{aligned}$$

(3) 成对密钥建立

节点 u 计算与 v 共享的成对密钥 $K_{uv} = f_{K_v^i}(u)$ ；节点 v 也按照相同方法计算其与 u 共享的成对密钥 K_{uv} 。

(4) 密钥删除

节点 u 的定时器经过时间 T_{\min} 后定时结束后，节点 u 删除 K_{IN}^i 和其所有相邻节点的主密钥 K_v^i 。但是，节点 u 不会删除自己的主密钥 K_u^i 或者其他预先加载的主密钥 K_u^j ， $i < j \leq M$ 。

例如，在第一个时间间隔 T_1 布置的节点 u 加载有 K_{IN}^1 和 K_u^j ， $1 < j \leq M$ 。节点 u 推导出自己的主密钥 K_u^1 ，并将 K_u^1 作为初始网络密钥用于建立与其相邻节点共享的成对密钥，经过时间 T_{\min} 后删除 K_{IN}^1 。网络控制器或者任务权威也应该删除 K_{IN}^1 （因为不再需要）。在第二个时间间隔 T_2 增加一个传感器节点 v 时，网络控制器给节点 v 加载初始密钥 K_{IN}^2 ，节点 v 根据 K_{IN}^2 推导出自己的主密钥 K_v^2 以及 $m-2$ 个主密钥 K_v^3 、 \dots 、 K_v^M 。节点 v 利用 K_{IN}^2 建立与其相邻节点（在时间间隔 T_1 或者 T_2 布置的）共享的成对密钥，这是因为节点 v 能够推导出其相邻节点在时间间隔 T_2 的主密钥。当节点 v 的定时器定时结束时，节点 v 删除 K_{IN}^2 。

安全分析：攻击者攻克在时间间隔 T_i 布置的一个新节点 u ，在时间 T_{\min} 内获得节点 u 的 K_{IN}^i 和 $M-i+1$ 个主密钥。节点 u 不知道任何初始密钥 K_{IN}^j （ $1 \leq j < i$ ），也不知道任何其他节点在 T_i 之前的时间间隔的主密钥，因此，攻击者不知道其他节点在过去时间间隔内建立的任何成对密钥。所以，这种对抗措施提供弱后向密钥机密性。此外，攻击者不能推导出随后时间间隔的初始密钥 K_{IN}^j （ $i < j \leq M$ ），所以无法知道其他节点在随后时间间隔中建立的成对密钥。因此，这种对抗措施提供弱前向密钥机密性。

假如攻击者能够在时间 T_{\min} 内攻克一个传感器节点，那么攻击者就能够进行节点增加攻击：给新节点加载正确的主密钥，然后将其引入到网络中。可以利用这些新节点来进行各种攻击，使传感器网络无法完成所承担的任务。防止这种攻击的一种简单方法是网络控制器在 T_{\min} 后广播新节点的 ID。节点检查每个新相邻节点的 ID，确定这个相邻节点是否合法。假如是不合法的相邻节点，则丢掉与这个失密相邻节点共享的成对密钥。这个方法存在两个问题：第一，假如节点 ID 是线性递增的，那么攻击者很容易推测出新节点的 ID；第二，假如需要增加很多节点，那么可能必须广播大量节点 ID（分组），消耗很多能量。这个问题的处理方法如下：假设网络控制器需要在时间间隔 T_i 给网络增加 N_i 个新节点，所以网络控制器根据随机种子 s_i 和

伪随机数生成器给这些新节点生成 N_i 个 ID，每个新节点加载一个唯一的 ID。完成布置后，每个新节点立即建立与其每个相邻节点共享的成对密钥。经过 T_{\min} 后，网络控制器给整个网络广播 N_i 和 s_i 。因此，早先布置的节点就能够验证一个新相邻节点的 ID 是否为根据 N_i 和 s_i 推导出来的一个 ID。假如节点 ID 足够长，那么攻击者很难伪造有效的 ID。

11.4.5 分群密钥的建立

每个节点拥有一个唯一的分群密钥，用于加密广播给自己群内成员的消息。分群密钥建立过程与成对密钥建立过程相同。考虑节点 u 需要建立一个分群密钥，其群内成员为所有直接相邻节点 v_1, v_2, \dots, v_m 。节点 u 首先生成一个随机密钥 K_u^c ，然后利用与每个相邻节点 v_i ($1 \leq i \leq m$) 共享的成对密钥加密 K_u^c ，接着将加密 K_u^c 发送给 v_i 。

$$u \rightarrow v_i: (K_u^c)_{K_{uv_i}}$$

每个节点 v_i 解密 K_u^c 并将 K_u^c 存储在表格中，然后使用其成对密钥加密自己的分群密钥后再发送给节点 u 。当节点 u 被排除时，每个相邻节点采用相同方法产生一个新的分群密钥，并将其发送给所有其他相邻节点。

11.4.6 全网密钥的建立

一个全网密钥是网络中所有节点共享的密钥。当网络控制器分发机密消息（如兴趣查询、指令）给网络中所有节点时就需要全网密钥。建立全网密钥的简单方法就是给每个节点预先加载全网密钥。但是由此出现的一个重要问题是：当检测到失密节点时，必须安全更新全网密钥。也就是说，必须改变全网密钥，并安全、可靠、及时地将新的全网密钥分发给剩余的所有节点。这等效于组密钥重新建立操作，这个组包含网络中的所有节点。

LEAP+以分群密钥为基础重新建立全网密钥。下面首先讨论认证节点的排除问题（这是密钥重新建立必需的），然后详细描述安全密钥分发机制。

1. 认证节点的排除

在传感器网络中，网络控制器广播的所有消息都必须加以认证；否则，外部攻击者就可以伪装网络控制器。因此，一个节点排除声明必须在其分发期间加以认证。

μ TESLA 协议效率高、能够容忍分组丢失，所以 LEAP+也采用 μ TESLA 协议进行广播认证。设被排除的节点为 u ，新的全网密钥为 K'_g ，被暴露的 μ TESLA 密钥为 K_i^T 。网络控制器广播如下消息 M ：

$$M: \text{Controller} \rightarrow *: u, f_{K'_g}(0), \text{MAC}(K_i^T, u | f_{K'_g}(0))$$

将 $f_{K'_g}(0)$ 称为确认密钥，因为节点利用 $f_{K'_g}(0)$ 能够验证其稍后将接收到的全网密钥 K'_g 的真实性。经过一个 μ TESLA 间隔时间后，网络控制器分发消息认证码 K_i^T 。又经过一个 μ TESLA 间隔时间后，节点 v 接收到消息 M 和消息认证码 K_i^T ，采用 μ TESLA 验证 M 的真实性。假如消息 M 通过验证，那么节点 v 临时存储确认密钥 $f_{K'_g}(0)$ 。最后，假如节点 v 是节点 u 的一个相邻节点，那么节点 v 删除其与节点 u 共享的成对密钥，并更新其分群密钥。

2. 安全密钥分发

全网密钥安全分发不要求使用特定路由协议。但是，为了具体起见，这里假定使用类似于 TinyOS 信标协议的路由协议。在这个路由协议中，在路由更新基础上将网络节点组织成宽度优先生成树，中心节点周期性广播路由更新并递推式地将其传播到网络中。通过生成树上递推传播过程将新全网密钥 K'_g 分发给所有合法传感器节点。中心节点利用其分群密钥加密 K'_g ，然后将其发送给生成树上的每个子节点。节点 v 接收到 K'_g 后能够立即验证 K'_g 的真实性：计算和检查 $f_{K'_g}(0)$ 是否等于其先前在节点排除消息中接收到的确认密钥。沿着生成树自上而下递推持续进行这个算法：接收到 K'_g 的每个节点 v 利用自己的分群密钥加密 K'_g ，然后将加密 K'_g 发送给其在生成树上的子节点。

尽管对加密广播消息进行逐跳解密涉及一定的计算开销，但是对于广播一个全网密钥不值得计较这些计算开销，这是因为只有一个密钥需要解密，并且密钥重建是相对较少发生的事件。

即使没有发生节点排除事件，也仍然需要周期性更新全网密钥。这对于对抗密码分析、防止攻击者在一个节点失密后、解密之前的所有广播消息非常重要。在 LEAP+ 中，网络控制器可以周期性广播一条认证密钥更新指令。假如网络节点松散时间同步，那么每个节点可以在固定时间间隔更新全网密钥：对全网密钥进行单向函数计算。每个节点可以推导出一个新的全网密钥 $K'_g = f_{K_g}(1)$ 来替换原来的全网密钥 K_g 。

11.4.7 本地广播认证

对于安全传感器网络，网络中的每条消息在转发或者处理之前必须加以认证，否则攻击者就可能向网络注入许多虚假分组而耗尽传感器节点的能量。这就要求认证方案非常简单，否则传感器节点就会一直不停地忙于验证注入的虚假分组。

本地广播是传感器网络一种很普遍的操作，本地广播消息（如路由控制消息、传感器累积感知数据）通常是事件驱动的或者时间驱动的，传感器节点通常无法事先知道其下一个发送分组的内容，从而妨碍直接使用 μ TESLA 进行本地广播认证，这是因为 μ TESLA 采用延迟暴露密钥技术。假如一个节点有 m 个相邻节点，则这个节点需要计算 m 个消息认证码，并将其封装到一条本地广播消息中，因而成对密钥不适用于本地广播认证。

为了支持本地广播认证，节点需要使用其所有相邻节点都知道的密钥，才只需要计算一个消息认证码，并将其封装到一条消息中。分群密钥满足这个要求，但是采用分群密钥进行本地广播认证易受伪装节点攻击。攻击者若是攻克一个传感器节点，就能够向网络注入虚假分组，同时假扮成一个相邻节点，使用这个相邻节点的分群密钥认证消息。由于分群密钥的对称性，所以不能击败这种假扮攻击。因此，LEAP+ 能最大程度地阻止这种假扮攻击。LEAP+ 本地广播认证方法是弱源认证方法，而 μ TESLA 是强广播源认证方法。

LEAP+ 本地广播认证的主要思想是使用一次性认证密钥，更明确地说是采用单向密钥链进行弱一跳广播认证。这种技术不同于 μ TESLA，没有采用延迟暴露密钥技术，也不要求相邻节点之间时间同步。每个传感器节点生成一个一定长度的单向密钥链，然后按照认证方式将这个密钥链的第一个密钥（第一个字节）发送给每个相邻节点。将节点单向密钥链中的一个密钥称为该节点的 AUTH 密钥。一个节点只要有消息需要发送，就将其密钥链中下一个 AUTH 密钥

封装到该消息中。按照 AUTH 密钥产生顺序的反顺序暴露 AUTH 密钥。接收相邻节点根据第一个密钥，或者最近从发送节点接收到的一个 AUTH 密钥就能够验证这条消息。

这个本地广播认证方法受到两个观察事实的激励，第一个观察事实是一个节点只需要认证给其直接相邻节点的一个分组（比如路由控制消息）；第二个观察事实是当一个节点发送一个分组时，其中一个相邻节点在接收到其他节点转发来的相同分组复制之前正常接收到这个分组。这是正确的，原因在于有关节点之间的距离的三角不等式，如图 11-4（a）所示，当节点 u 发送一个分组（包含 M 和 AUTH 密钥）时，由于 $|uv| < |ux| + |xv|$ ，所以节点 v 在接收到节点 x 转发来的相同分组拷贝之前接收到这个分组。因此，一旦节点 v 接收到消息 M ，那么攻击者 x 在假扮成节点 u 时就不能重复使用这个 AUTH 密钥来注入另一个分组。

上述本地广播认证方法提供弱源认证，同时允许被动参与。但是，攻击者可以突破这种本地广播认证：防止节点 v 直接接收节点 u 发送的分组。例如在图 11-4（b）中，攻击者让另一个节点 w 在节点 u 发送时对节点 v 进行发送，从而屏蔽或者干扰节点 v 。随后攻击者假扮成节点 u ，给节点 v 发送篡改过的分组。节点 v 接收不到具有相同 AUTH 密钥的分组，因此将会接收被篡改的分组。

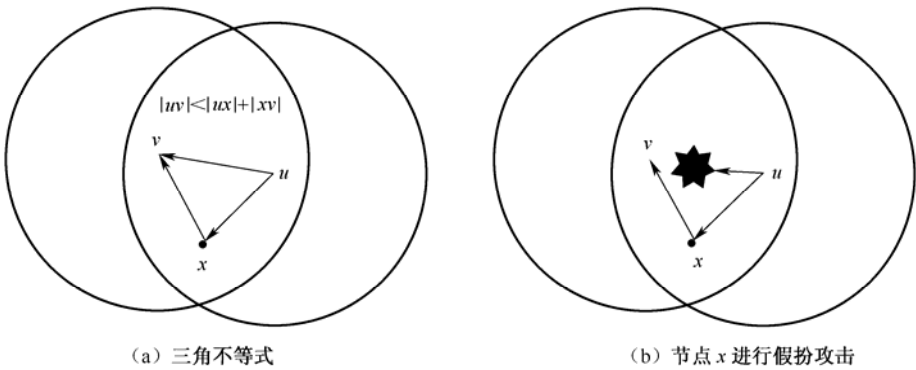


图 11-4 三角不等式和假扮攻击

利用分群密钥加密 AUTH 密钥，就能够防止外部攻击者进行上述攻击。外部攻击者不知道节点 u 的分群密钥，因此不能假扮成节点 u 。内部攻击者知道节点 u 的分群密钥，上述本地认证方法不能对抗内部攻击者的攻击。但是应该注意到：①由于散列函数的单向特性，失密节点 x 假扮成节点 u 能够注入到网络中的最大虚假分组数量受到节点 u 已经发送的分组数量的限制；②由于传感器节点 x 只有其一跳相邻节点的分群密钥，所以节点 x 的失密只允许攻击者在以节点 x 为中心的两跳区域内进行攻击。

11.4.8 LEAP+安全分析

下面分析 LEAP+各个密钥机制的安全性能，首先分析失密节点未被发现时网络的顽存能力，然后分析 LEAP+对抗各种路由攻击的强壮性。

1. 顽存能力

当一个传感器节点 u 被攻失密后，攻击者可以利用节点 u 的密钥信息进行攻击。假如检

测出失密事件，那么 LEAP+密钥重新建立方案能够有效从网络中排除失密节点 u 。失密节点被排除后，攻击者不能再进行攻击。

传感器网络经常布置在无人照看的环境中，所以其中的失密事件检测比较困难。因此，传感器网络在面对失密节点未被检测出来时的顽存能力是传感器网络最重要的安全要求之一。下面分析 LEAP+的顽存能力。

假如每个节点利用其单独密钥直接向中心节点报告其感知数据，那么获取单独密钥就允许失密节点注入虚假感知数据。攻击者拥有失密节点的成对密钥和分群密钥，就能够建立与失密节点的所有相邻节点的信任，从而能够对网络注入一些恶意路由控制信息和错误的传感器感知数据。但是在 LEAP+中，由于采用一次性密钥进行本地广播认证，所以攻击者通常必须利用失密节点的身份进行这种攻击。LEAP+的一个显著特征就是具有安全受损范围本地化能力：传感器网络布置完毕后，每个传感器节点维护一张可信任相邻节点表，因此失密节点不能建立与其相邻节点之外任何节点的信任关系，也不能危害其他节点之间的链路安全。

攻击者拥有全网密钥，就能够解密中心节点广播的消息。因为中心节点的广播消息是希望每个传感器节点都能接收的，所以单独一个传感器节点被攻失密后就足够解密中心节点的广播消息，而不用关心消息安全分发的安全机制。但是，攻击者即使拥有全网密钥，也仍然不能假扮成中心节点进行全网恶意分组泛洪，这是因为中心节点发送的任何消息都必须通过 μ TESLA 的认证。由于全网密钥是周期性更新的，所以攻击者只能利用其当前拥有的全网密钥解密经这个当前全网密钥加密的消息。

2. 安全路由抗攻击能力

在 LEAP+中，采用本地广播认证方法认证路由控制信息，本地广播认证能够防止大多数外部攻击，但是不能防止蠕虫攻击。因此，下面主要分析已经攻克一个或者多个传感器节点的内部攻击者进行的攻击。

内部攻击者可能进行路由信息的哄骗、篡改、重放，达到以下目的：产生路由闭环、吸引或者抵制网络流量、产生错误消息。内部攻击者也可以进行选择性转发攻击，失密节点抑制几个经过精心选择的节点产生的路由分组，同时可靠转发其余分组。LEAP+不能防止攻击者进行选择性转发攻击，但是能够挫败选择性转发攻击效果或者使攻击效果最差。第一，LEAP+本地广播认证方法使得选择性转发攻击只可能在失密节点的一跳范围内进行；第二，由于攻击者攻击范围被限制在一个局部小范围内，所以攻击者进行选择性转发攻击时被检测出来的概率高，因为失密节点转发的篡改消息可能被发送节点所旁听到，所以篡改攻击也很可能被检测出来；第三，失密节点一旦被检测出来，LEAP+密钥重建方法就能够非常有效地从网络中排除这个失密节点。

LEAP+能够很大程度上防止 hello 泛洪攻击。攻击者进行 hello 泛洪攻击时，采用足够高的发射功率将 hello 消息发送给所有网络节点，使所有网络节点相信攻击者是其相邻节点。WSN 的很多路由协议（如定向扩散、LEACH、SPAN、TinyOS 信标）都易受 hello 泛洪攻击。假如攻击者成功进行 hello 泛洪攻击，那么所有网络节点发送的感知数据或者其他分组将被淹没。但是在 LEAP+，除了相邻节点寻找阶段，攻击者在其他时间不能成功进行 hello 泛洪攻击，这是因为每个节点只接收其通过认证的相邻节点发送的分组。

3. 蠕虫攻击和污水池攻击的对抗能力

极难检测的攻击是蠕虫和污水池的联合攻击。在污水池攻击中，失密节点广播诸如高剩余能量、端到端高可靠性之类的信息，吸引其相邻节点发送的分组（如感知数据），然后却丢掉这些分组。这些信息很难验证。在蠕虫攻击中，通常两个相距很远的恶意节点具有一条额外低时延链路，相互转发从其相邻节点旁听到的分组；攻击者将其中一个恶意节点布置在中心节点附近、另一个恶意节点布置在兴趣目标附近（兴趣目标附近的节点离中心节点的实际距离多达数个转发跳），就能够使这些节点相信自己离中心节点的距离只有一个或者两个转发跳，从而产生一个污水池。类似地，由于蠕虫攻击，实际相距数个转发跳的节点可能认为是相邻节点。攻击者不需要攻克任何传感器节点就能够进行蠕虫攻击，因此蠕虫攻击非常有效。

在 LEAP+ 中，除了在成对密钥建立过程的相邻节点寻找期间，外部攻击者不能成功进行蠕虫攻击。经过相邻节点寻找期间后，节点知道其所有相邻节点；因此攻击者随后不能使相距很远的两个节点相信自己是对方的相邻节点。由于相邻节点寻找所需时间很短（几十秒），所以攻击者进行蠕虫攻击的成功概率非常小。通过认证的相邻区域信息对于防护蠕虫攻击非常关键。

在 LEAP+ 中，内部攻击者至少需要攻克两个传感器节点才能进行蠕虫攻击，即使至少两个传感器节点由于内部攻击者攻击而失密，但是内部攻击者仍然不能使相距很远的两个节点完成其相邻节点寻找之后相信自己是对方的相邻节点。但是，假如中心节点附近一个节点 u 受攻失密、同时兴趣区域内另一个节点 v 也受攻失密，那么内部攻击者可能能够成功使节点 v 成为一个污水池，这是因为节点 v 和中心节点之间的距离变短，从而使得节点 v 对周围节点特别有吸引力。在中心节点位置固定不变的传感器网络应用中，网络拓扑构成后，节点大约知道其到达中心节点的转发跳距离，因此攻击者很难产生不会被检测出来、同时极有吸引力的污水池。

11.4.9 LEAP+性能评估

下面分析分群密钥和全网密钥的更新开销。在分析时假定 LEAP+ 密钥重建协议使用生成树将新的全网密钥交付给所有网络节点。

1. 计算开销

当更新一个分群密钥时，被排除节点的一个相邻节点必须利用与其每个相邻节点共享的成对密钥加密新分群密钥。因此，这种加密次数由相邻节点数决定，而相邻节点数依赖传感器网络的节点密度。设被排除节点有 s 个相邻节点数，其中每个相邻节点有 d_i 个合法相邻节点， $1 \leq i \leq s$ ；因此总加密次数 $X_e = \sum_{i=1}^s d_i$ ；尽管这 s 个相邻节点中的一个节点的相邻节点的解密次数依赖其所在位置，但是总解密次数仍然等于 $\sum_{i=1}^s d_i$ 。在最坏情形（即一个节点是所有这 s 个相邻节点的相邻节点）下，需要解密 s 个密钥。对于单个节点，总共执行的对称密钥操作次数局限于 $[\max(d_i) + s - 1]$ 。对于 N 个传感器节点的网络（包括被排除节点），一个节点在更

新分群密钥时平均执行的对称密钥操作次数等于 $2X_e/(N-1)=2\sum_{i=1}^S d_i/(N-1)$ 。

在安全分发全网密钥期间，每个节点解密一个密钥，所以总解密次数等于 $N-1$ 。LEAP+ 使用分群密钥转发全网密钥，因此父节点只需要对其所有子节点加密一次。总加密次数依赖网络拓扑，最大等于 $N-1$ 。因此，对称密钥操作总次数最大等于 $2(N-1)$ ，平均每个节点的对称密钥操作次数最大等于 2。

上述分析表明：排除一个节点的计算开销由网络密度决定。在 N 个传感器节点的网络中，每个节点的节点密度为 d ，当求所有计算开销之和时，平均每个节点的对称密钥操作次数约等于 $2\sum_{i=1}^{d-1}(d-1)/(N-1)+2=2(d-1)^2/(N-1)+2$ 。对于密度合理的网络，计算开销不会成为 LEAP+ 的性能瓶颈。例如，对于 $N=1\ 000$ 、节点密度等于 20 的一个传感器网络，平均计算开销是：每次排除一个节点时，平均每个节点的对称密钥操作次数等于 2.7。 N 越大，平均计算开销越低。

2. 通信开销

重新建立全网密钥的通信开销分析类似于计算开销的分析。对于由于节点排除而更新分群密钥，对于一个 N 个传感器节点、节点密度等于 d 的传感器网络，一个节点发送和接收的平均密钥数量等于 $(d-1)^2/(N-1)$ ，因此，每个节点平均发送和接收的密钥数随着网络节点密度 d 的增大而增大，但是随着网络规模 N 的增大而减小。。在全网密钥安全分发期间，一个节点发送和接收的平均密钥数等于 1。例如，对于 $N=1\ 000$ 、节点密度等于 20 的一个传感器网络，每次排除一个节点时，每个节点平均发送和接收 1.4 个密钥。

3. 存储要求

在 LEAP+ 中，每个节点必须维持四种密钥。假如一个节点有 d 个相邻节点，那么这个节点必须存储 1 个单独密钥、 d 个成对密钥、 d 个分群密钥、1 个全网密钥。此外，在 LEAP+ 本地广播认证中，每个节点还要维持自己的单向密钥链以及第一个密钥、每个相邻节点的最近 AUTH 密钥。传感器网络的分组发送速率通常很低。例如，周期性产生和转发传感器感知数据，偶尔交换路由控制信息。因此，传感器节点能够存储长度合理的一个密钥链。密钥链中的密钥用完后，节点会产生和引导一个新的密钥链。假如网络流量载荷很重，那么最好使用长密钥链。为了避免存储长度等于 n 的整个密钥链，LEAP+ 采用一种优化算法^[4]：为每个 AUTH 密钥计算 $O(\text{lb}\sqrt{n})$ 个散表，并使用 $O(\text{lb}\sqrt{n})$ 密钥。每个节点为其密钥链存储 L 个密钥。因此，每个节点存储的密钥数等于 $3d+2+L$ 。例如，当 $d=20$ 、 $L=30$ 时，假如一个密钥长 8 B，那么一个节点存储 92 个密钥（736 B）。

总之，就计算、通信、存储要求而论，LEAP+ 是可扩展、高效的协议。

参 考 文 献

[1] National Institute of Standards and Technology (NIST), DES model of operation, Federal Information Processing Standards Publication 81(FIPS PUB 81) (1981).

- [2] R.L. Rivest. The RC5 encryption algorithm. in: Workshop on Fast Software Encryption (1995) pp.86-96.
- [3] ANDERSON, R. AND KUHN, M. Tamper resistance—a cautionary note. In Proceedings of the 2nd USENIX Workshop on Electronic Commerce'96, pp.1–11,1996..
- [4] COPPERSMITH, D. AND JAKOBSSON,M. Almost optimal hash sequence traversal. In Proceedings of Finanical Cryptography (FC'02), pp.102–119, 2002.
- [5] DENG, J., HARTUNG, C., HAN, R., AND MISHRA, S. A practical study of transitory master key establishment for wireless sensor networks. In Proceedings of the 1st IEEE/CreateNet Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm), pp.289–299, 2005.
- [6] HU, Y., JAKOBSSON, M., AND PERRIG, A. Efficient constructions for one-way hash chains. In Proceedings of Applied Cryptography and Network Security (ACNS), 2005.
- [7] LIU, D. AND NING, P. Establishing pairwise keys in distributed sensor networks. In Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS'03),pp.52–61,2003.
- [8] PARNO, B., PERRIG, A., AND GLIGOR, V. Distributed detection of node replication attacks in sensor networks. In Proceedings of the IEEE Symposium on Security and Privacy,2005.
- [9] H. Chan and A. Perrig. Security and Privacy in Sensor Net works. IEEE Comp. Mag., Oct. 2003, pp.103–05.
- [10] E. Shi and A. Perrig. Designing Secure Sensor Networks. Wireless Commun. Mag., vol.11, no.6, Dec. 2004, pp.38–43.
- [11] A. Perrig et al.. SPINS: Security Protocols for Sensor Networks. Wireless Networks, vol.8, no.5, Sept. 2002, pp.521–34.
- [12] L. Hu and D. Evans. Secure Aggregation for Wireless Networks. Wksp. Security and Assurance in Ad Hoc Networks, 2003.
- [13] B. Przydatek, D. Song, and A. Perrig. SIA: Secure Information Aggregation in Sensor Networks. SenSys '03: Proc. 1st Int'l. Conf. Embedded Networked Sensor Systems, New York: ACM Press, 2003, pp.255–265.
- [14] S. Zhu et al.. An Interleaved Hop-by-Hop Authentication Scheme for Filtering of Injected False Data in Sensor Networks. Proc. IEEE Symp. Security and Privacy, Oakland, CA, May 2004, pp.259–271.
- [15] B. Deb, S. Bhatnagar, and B. Nath. Information Assurance in Sensor Networks. Proc. 2nd ACM Int'l. Conf. Wireless Sensor Networks and Applications (WSNA '03), New York: ACM Press, 2003, pp.160–168.
- [16] C. Karlof and D. Wagner. Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures. Proc. First IEEE Int'l. Wksp. Sensor Network Protocols and Applications, May 2003, pp.113–27.
- [17] J. Newsome et al.. The Sybil Attack in Sensor Networks: Analysis and Defenses. IPSN '04: Proc. IEEE Int'l. Conf. Info. Processing in Sensor Networks, Apr. 2004.
- [18] Y.-C. Hu, A. Perrig, and D. B. Johnson. Packet Leashes: A Defense Against Wormhole

- Attacks in Wireless Networks. Proc. IEEE INFOCOM 2003, Apr. 2003.
- [19] R. Watro et al.. TinyPK: Securing Sensor Networks with Public Key Technology. SASN '04: Proc. 2nd ACM Wksp. Security of Ad Hoc and Sensor Networks, New York: ACM Press, 2004, pp.59–64.
 - [20] C. Karlof, N. Sastry, and D. Wagner. TinySec: A Link-Layer Security Architecture for Wireless Sensor Networks. SenSys'04: Proc. 2nd Int'l. Conf. Embedded Networked Sensor Systems, New York: ACM Press, 2004, pp.162–75.
 - [21] S. A. Camtepe and B. Yener. Key Distribution Mechanisms for Wireless Sensor Networks: A Survey. Computer Science Department at RPI, Tech. Rep. TR-05-07, 2005.
 - [22] S. Zhu, S. Setia, and S. Jajodia. LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks. CCS'03: Proc. 10th ACM Conf. Comp. and Commun. Security, New York: ACM Press, 2003, pp.62–72.
 - [23] L. Eschenauer and V. D. Gligor. A Key-Management Scheme for Distributed Sensor Networks. CCS '02: Proc. 9th ACM Conf. Comp. and Commun. Security, New York: ACM Press, 2002, pp.41–47.
 - [24] H. Chan, A. Perrig, and D. Song. Random Key Predistribution Schemes for Sensor Networks. Proc. IEEE Symp. Security and Privacy, May 2003.
 - [25] D. Liu and P. Ning. Establishing Pairwise Keys Distributed Sensor Networks. CCS '03: Proc. 10th ACM Conf. Comp. and Commun. Security, New York: ACM Press, 2003, pp.52–61.
 - [26] R. D. Pietro, L. V. Mancini, and A. Mei. Random Key-Assignment for Secure Wireless Sensor Networks. SASN '03: Proc. 1st ACM Wksp. Security of Ad Hoc and Sensor Networks, New York: ACM Press, 2003, pp.62–71.
 - [27] W. Du et al.. A Pairwise Key Predistribution Scheme for Wireless Sensor Networks. CCS '03: Proc. 10th ACM Conf. Comp. and Communications Security, New York: ACM Press, 2003, pp.42–51.
 - [28] W. Du et al.. A Key Management Scheme for Wireless Sensor Networks using Deployment Knowledge. Proc. IEEE INFOCOM, Hong Kong, 2004, pp. 586–97.
 - [29] D. Liu and P. Ning. Multilevel μ TESLA: Broadcast Authentication for Distributed Sensor Networks. Trans. Embedded Computing Sys., vol. 3, no. 4, 2004, pp. 800–36.
 - [30] D. Liu et al.. Practical Broadcast Authentication Sensor networks. MobiQuitous '05: Proc. 2nd Annual Int'l. Conf. Mobile and Ubiquitous Systems: Networking and Services, July 2005, pp.118–29.
 - [31] W. Du, R. Wang, and P. Ning. An Efficient Scheme for Authenticating Public Keys Sensor Networks. MobiHoc '05: Proc. 6th ACM Int'l. Symp. Mobile Ad Hoc Net. and Comp., New York: ACM Press, 2005, pp.58–67.
 - [32] J. Deng, R. Han, and S. Mishra. Security Support for in-Network Processing Wireless Sensor Networks. SASN '03: Proc. 1st ACM Wksp. Security of ad Hoc and Sensor Networks, New York: ACM Press, 2003, pp.83–93.
 - [33] H. Çam, D. Muthuavinashiappan, and P. Nair. ESPDA: Energy Efficient and Secure Pattern-Based Data Aggregation for Wireless Sensor Networks. Proc. IEEE Sensors, Toronto,

Canada, Oct. 2003, pp.732–36.

- [34] D. Wagner. Resilient Aggregation Sensor Networks. SASN'04: Proc. 2nd ACM Wksp. Security of Ad Hoc and Sensor Networks, New York: ACM Press, 2004, pp.78–87.
- [35] S. Zhu, S. Setia, and S. Jajodia. LEAP+: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks. Proc. ACM Transactions on Sensor Networks, Vol.2, No.4, November 2006, pp.500–528.

第 12 章 无线传感器网络中间件技术

在传统环境（建立了良好的操作系统功能）中，中间件常常位于操作系统与应用之间。但是在 WSN 中，操作系统接口仍然是一个值得研究的问题，很多应用没有操作系统组件而直接执行硬件操作。通常，传感器网络中间件可以定义为提供数据累积、自适应目标应用需求的控制与管理机制的软件，从传感器网络中收集数据。

12.1 WSN中间件面临的挑战

诸如目标检测、战场监视、反恐之类的军事应用最早推动了传感器网络应用。但是，传感器网络相对于传统网络的优势使得传感器网络具有很多其他潜在的应用，包括从基础设施安全到工业感知，如环境与栖息地监视、医疗卫生应用、家庭自动化、交通控制等。成功设计和开发一个 WSN 中间件层必须同时考虑 WSN 特点和应用引起的许多挑战。

① 有限能量与资源的管理：微电子技术的进步使得人们能够设计只有一个立方厘米的微型装置。这种微小装置的能量有限、单独资源（如 CPU 和存储器）有限。将数百个甚至数千个这种微小装置布置在恶劣、敌对环境中，在这种情况下很难甚至不可能依靠人工替换和维护装置，无线媒介是唯一的远程访问方法。因此，中间件应该提供处理器和存储器的高效使用机制以及低功率通信机制。一个传感器节点应该完成三种基本操作——感知、数据处理、通信，但是不会耗尽资源。例如，在能量意识中间件中，根据应用很可能装置的大部分组件（包括电台）在大部分时间被关电。

② 可扩展性、移动性、动态网络拓扑：假如一个应用扩大，那么网络应该足够灵活，随时随地允许这种扩大而不会影响网络性能。随着网络的增大，有效中间件服务必须能够维持可接受的性能。网络拓扑易受诸如故障、装置失效、移动障碍物、移动、干扰之类的因素影响而频繁变化。不论传感器网络动态性如何变化，中间件都应该支持传感器网络强壮操作、自适应网络环境的不断变化，还应该支持容错机制和传感器节点自构自维护机制。

③ 异类性：中间件应该提供低级编程模型，以满足弥补硬件技术本身能力以及必需的广泛活动（诸如重构、执行、通信等）之间缺口所面临的挑战。中间件应该建立系统机制，与各种类型硬件和网络接口，这些系统机制只得到分布式、简单操作系统抽象的支持。

④ 动态网络结构：传感器网络必须处理动态资源（如能量、带宽、处理能力），必须支持长时间运行的应用，因此必须设计高效的传输协议、路由协议、MAC 协议，使网络尽可能长时间运行。因为对网络的掌握对于网络正确操作是必要的，所以中间件应该提供 Ad Hoc 网络资源寻找功能。传感器节点需要知道自己在网络及其整个网络拓扑中的位置，在有些情况下，采用 GPS 自定位不可能、不可靠或者费用昂贵。重要系统参数（如网络大小、每平方英里的节点密度等）影响时延、可靠性、能量之间的综合平衡。

⑤ 真实世界综合：大多数传感器网络应用都是实时现象，其时间和空间极其重要。因此，中间件应该提供实时服务和一致性数据，适应现象的实时变化。

⑥ 应用认知：应用认识设计原则决定了 WSN 中间件另一个重要而独特的属性。中间件必须包含应用对 WSN 基础设施认识的注入机制，使应用开发人员将应用通信要求映射为网络参数，从而调整网络监视。许多现有中间件与特定应用结合在一起。但是，中间件应该支持广泛的应用。所以，开发人员必须研究应用专一性与中间件通用性之间的平衡。

⑦ 数据累积：大多数传感器网络应用涉及特定区域内且包含冗余数据的节点。这些特点使得能够对来自不同源的数据进行网内累积，排除冗余数据，使对中心节点的发送最少。通信开销比计算开销高很多，因此网内累积能够节省相当大一部分能量和资源。累积将重点从传统的地址中心网络法转移到数据中心网络法。

⑧ 服务质量：不同的研究团体和技术团体对服务质量的定义不同。在 WSN 中，从特定应用和网络两个方面认识服务质量：特定应用 QoS 是针对特定应用的 QoS 参数（如传感器节点度量、布置、覆盖范围、活动传感器节点数量）；网络 QoS 是如何支持通信网络才能够满足应用需求，同时高效使用网络资源（如带宽、能量、存储器、功耗等）。由于 WSN 资源有限、资源动态性、网络拓扑动态性以及无线传输的固有缺陷等原因，有线网络的 QoS 机制不适合 WSN。因此，WSN 中间件应该提供新机制，维护长时间的 QoS，甚至在所需 QoS 和应用状态发生变化后调整 QoS 本身。WSN 中间件应该根据各种性能（如网络吞吐量、数据交付时延、能耗等）之间的综合平衡实现最佳设计。

⑨ 安全：WSN 正在广泛应用于各种领域，涉及敏感信息，比如卫生健康和营救。在恶劣环境中布置大规模无绳 WSN 使 WSN 暴露在恶意入侵和攻击（如拒绝服务）之中。此外，无线传输媒介易于偷听和注入敌方分组，危害网络功能。所有这些因素使得安全极其重要。传感器节点能量和处理能力均有限，所以标准安全机制（实现复杂、实现代码量大、资源消耗大）不适用于 WSN。这些挑战推动着和迫切需要开发全面而安全的解决方案，既实现较宽范围保护，又维持所需要的网络性能。在设计和开发中间件软件的开始阶段，就应该开发和综合安全功能，实现各种安全要求（如机密性、认证、完整性、信息新鲜、有效性等）。

12.2 WSN中间件的功能要求

传感器网络中间件的主要功能是支持基于感知应用的开发、维护、部署、执行，包括明确的复杂高级感知任务表述机制、将感知任务交付给 WSN 的机制、传感器节点协调机制（用于任务分解以及将（子）任务分散到各个单个传感器节点）、数据融合机制（将单个传感器节点的感知数据合并成高级数据结果）、数据结果报告机制（将融合得到的数据结果报告给中心节点），此外还应该提供处理异种传感器节点的合适抽象和机制，提供一种通用技术来从外部应用访问传感器数据，形成面向服务、自上而下、连接外部应用的标准体系架构。

下面详细介绍基于事件的 WSN 中间件 Impala、采用数据驱动法的中间件 SINA、面向 QoS 和服务的中间件 MiLAN。

12.3 ZebraNet系统中的中间件系统（Impala）

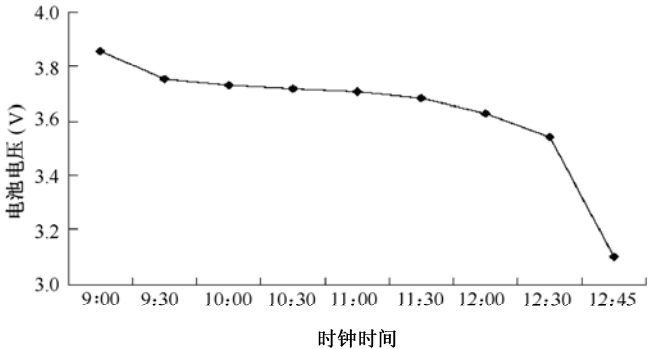
美国普林斯顿大学 ZebraNet 研究计划是开发能量高效移动传感器网络，用于辅助非洲斑马行走的跟踪。在肯尼亚 Mpala 研究中心附近的斑马脖子上安装一个环，环上配有所开发的

ZebraNet 传感器节点, 如图 12-1 (a) 所示。ZebraNet 环由白色丁基合成橡胶带材料组成, 环上较黑的点是太阳能模块 (位于两层丁基合成橡胶带中间), 胶带上的口子是为了让阳光照射内部的太阳能模块。每个 ZebraNet 传感器节点配备一个 GPS 单元, 用于确定和记录斑马的位置信息。采用对等协议使斑马位置信息从一个斑马传递到另一个斑马, 最终传递到达中心节点, 中心节点再处理、分析斑马位置信息。

Impala 系统是 ZebraNet 研究计划的一个组成部分, 是 ZebraNet 系统的中间件层, 支持应用模块化、简单、自适应性、可修复性。Impala 层作为一个操作系统、资源管理器、事件过滤器, 在其上面安装、运行特定应用。



(a) 佩戴 ZebraNet 环的草原斑马



(b) 电台持续数小时发送的功耗

图 12-1 佩戴 ZebraNet 环的草原斑马、电台持续数小时发送的功耗

12.3.1 ZebraNet系统简介

ZebraNet 是一个移动传感器网络, 其目标是采用能量高效跟踪节点和对等通信技术提高跟踪技术。ZebraNet 的近期研究重点是采用极少的基础设施进行大区域野生动植物跟踪, 而主要研究目标是大量固定和移动传感器的布置、管理、通信问题。

像 ZebraNet 之类的传感器网络涉及数十个、数百个计算装置的协调问题, 所以是一种并行系统。尽管自治、自适应计算是当今许多系统的共同研究问题, 但是 Impala 的研究动因直接来源于 ZebraNet 的预期使用, 其目标是布置 30 个或者更多的 ZebraNet 节点, 用于长期跟踪动物。随着研究的不断进行, 需要更新软件和自适应处理。但是, 由于计算节点安装在野生动物身体 (脖颈) 上, 需要野生动物安静时才能够取回节点和手工修改其软件, 因此手工软件升级很困难。

ZebraNet 硬件如图 12-2 所示, 主要组成模块包括微型控制器、GPS、外部 Flash 存储器、电台、太阳能充电电池。TI 公司的超低功率 MSP430F149 16 bit 微型控制器用于控制硬件。该处理器包含片内 2 KB RAM 和 64 KB Flash 存储器, 以及两个串行接口, 采用双时钟配置, 当访问感知外设、通信外设、外部存储器时按照 8 MHz 时钟运行, 在其他时候按照 32 kHz 时钟运行。32 kHz 时钟的功耗是 8 MHz 时钟功耗的一半, 按照 32 kHz 时钟运行时可替代处理器进入休眠。

μ -blox GPS-MS1E 体积小, 锁相速度快, 其典型的热启动捕获时间是 2~6 s。在 Impala 实验中, 捕获一个精确位置方位需 10~20 s。GPS 在微型处理器的一个串行端口上按照速率

38 400 波特率（该芯片的最大串行速率）的异步串行连接方式与微型处理器通信，GPS 与外部 Flash 存储器共享微型处理器这个串行端口。GPS 工作电源 3.3 V，可软件开/关。关闭 GPS 3.3 V 电源可节省能量。

选用 ATMEL 4 Mbit AT45DB041B 数据闪存芯片存储数据。在 ZebraNet 中，一个节点有足够存储空间存储自己 25 天的位置数据以及从其他节点接收到的 52 斑马天的位置数据。该 Flash 存储器芯片以 667 kb/s 速率与微型处理器同步通信，并且与 GPS 共享微型处理器的同一个串行接口，因此可以同时进行外部 Flash 和电台操作。外部 Flash 和微型处理器均需要连续加电。

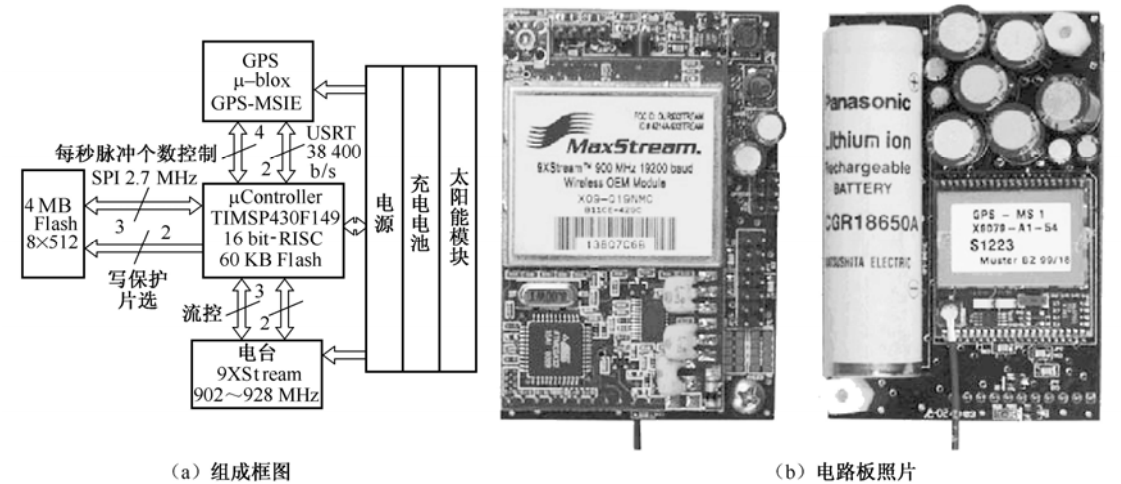


图 12-2 ZebraNet 节点

节点之间的无线数据传输采用 MaxStream 9XStream 电台来完成。电台工作频率 900 MHz，传输距离 5 英里，发射功率 1 W，接收灵敏度-107 dBm。电台在微型处理器的另一个串行端口上按照速率 38 400 波特率（与 GPS 速率匹配）的异步串行连接方式与微型处理器通信，与其他节点通信的空中传输速率为 19 200 b/s。但是在 ZebraNet 配置中，电台可靠传输距离 0.5~1 英里。电台工作电源+5 V，可软件开/关。关闭电台+5 V 电源可节省能量。

选用的供电电池是松下 CGR18650A2A 小时锂离子电池。电池充满电时 4.2 V，供电失效时 3.1 V。因此选择 3.4 V 作为电池工作电压的下限，这是因为电池电压为 3.4 V 时电台和 GPS 迅速耗尽电池能量，结果不能正常工作。电池可以使用太阳能电池[精心安置在斑马环上，见图 12-1 (a)]反复充电。

由于能量效率在移动传感器网络中很关键，所以 ZebraNet 硬件特点是低功率元件、高效电源。测试系统在一个循环（执行所有操作）中的功耗：给实验电路板[见图 12-2 (b)]加 4.0 V 电源，测试结果如表 12-1 所示。图 12-1 (b) 给出了电台持续发送的功耗。

ZebraNet 并不依靠对中心节点的持续访问，而是采用周期性节点寻找和对等通信、依靠其他对等节点的存储转发路由将数据传递给中心节点。节点不是采取面向连接的方法确定一条到达中心节点的完整路由，而是采用引导数据朝中心节点传递的试探法，通过其他节点的逐跳转发，将数据发送给中心节点。

在 ZebraNet 的第一个版本中，在每个节点上运行的主要“应用”软件是通信协议软件，

用于将数据传递给中心节点。在今后的传感器系统中，较复杂的应用软件包括传感器数据的过滤与融合软件以及通信软件。Impala 支持广泛的应用。

表 12-1 ZebraNet 节点硬件的功耗（工作电压 4.0 V）

方 式	电流/mA	功率/mW
CPU（工作时钟 32 kHz）	2.40	9.6
CPU（工作时钟 8 MHz）	4.83	19.32
GPS	142	568
电台发射	195	780
电台接收	78.1	312.4

12.3.2 ZebraNet中间件体系结构

许多传感器网络布置在恶劣环境中，网络运行数月甚至数年没有用户干预。有些传感器网络由数百个甚至数千个节点组成。有些传感器网络分布在极宽阔的地理区域内。因此，设计人员必须将传感器应用软件的长期管理作为设计过程的一个完整组成部分。通过采用能够动态更新和适应应用的中间件层，可以随时插入新协议，以及能够在各个协议之间任意切换。

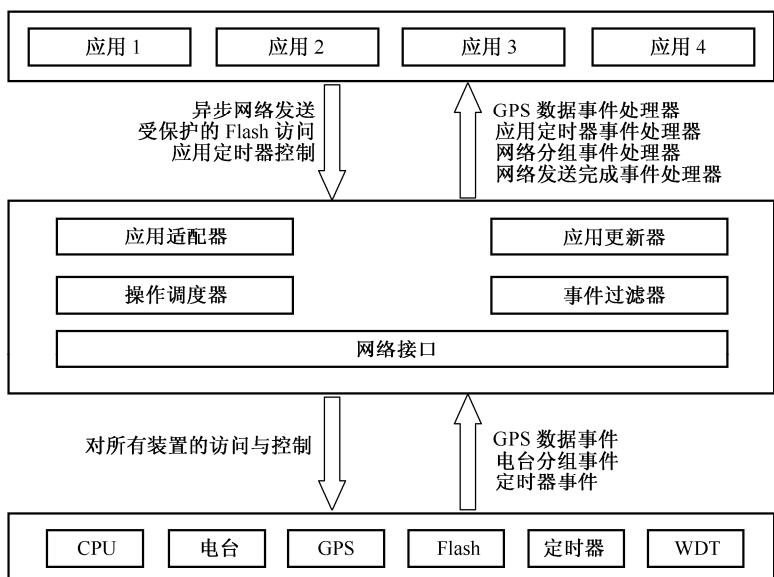
1. 基本设计原理

尽管将多个应用协议组合成一个自适应、自我更新的综合性协议是可能的，但是 Impala 分层法相对于单一法（Monolithic Approach）具有以下几个优点：

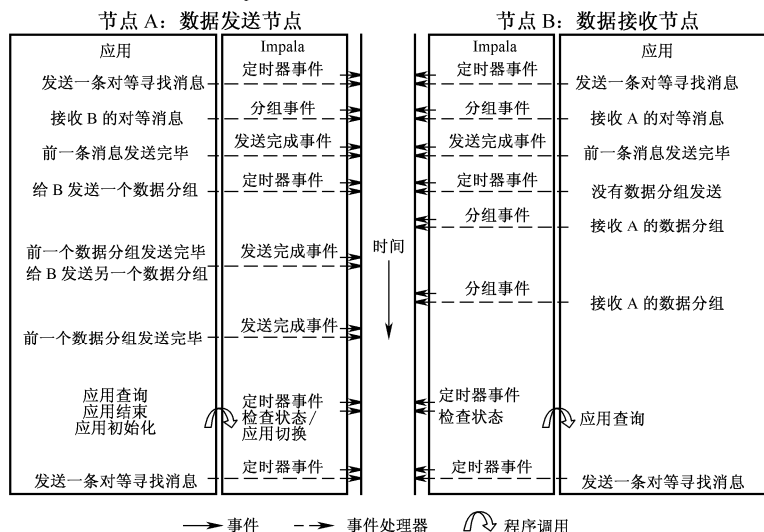
- ① 模块性：采用 Impala 中间件层处理切换决策，各个应用相互独立，不需要相互协调。还可以采用 Impala 中间件层处理更新问题，各个应用能够将其重点放在其目标上，可以不理睬同时性的软件执行与更新需求。
- ② 正确性：Impala 使应用正确性更易于实现，这是因为单个应用编程比超级应用编程简单，后者需要很多交互组件和更新组件。
- ③ 易于更新：诸如增加一个应用、撤销一个应用、修改一个应用之类的软件更改比较简单，这是因为软件更改只涉及一个模块内的本地代码变化。而在单一法中，即使很小的软件更改也可能影响到整个软件代码。
- ④ 能量效率：在进行软件更新时，Impala 不是发送一个完整的集成程序，而是发送较小的程序模块。因为在能量有限的传感器网络中发射机是耗能最多的组件，所以 Impala 节能明显强于单一法。

2. 系统体系结构

图 12-3（a）表示 Impala 的系统体系结构。Impala 有三层：最高层是应用层，接下来依次是 Impala 层和固件层。层与层之间的主要接口就是服务和事件。固件层通过服务接口向 Impala 输出许多硬件访问和控制功能。Impala 层防止这些固件功能被应用层直接使用，只按照裁减方式或者保护方式输出应用所需要的功能，另外还向应用层输出自己的网络接口。



(a) Impala 系统体系结构的分层与接口



(b) 基于事件的编程模型的时序例子

图 12-3 分层与接口、编程模型的时序例子

应用层包含所有应用协议和 ZebraNet 程序。各个应用使用各种策略完成一个公共任务：收集环境信息并采用对等传输技术将所收信息传递给中心节点。每次只运行一个应用。Impala 层除了包含操作调度器和网络接口之外，还包含三个中间件代理：应用适配器、应用更新器和事件过滤器。应用适配器使应用协议适应不同运行条件，以提高性能、能量效率、强壮性。应用更新器通过无线收发信机接收和传播软件更新，并将其安装在节点上。事件过滤器捕获事件，并将事件分发给顶层系统单元，初始化一系列处理。

Impala 有以下五类消息：

① 定时器事件：表示一个定时器定时结束的信号。Impala 有三个定时器分别属于当前活动应用、应用适配器、应用更新器。定时器拥有者处理自己的定时器事件。

② 分组事件：表示一个网络分组已经传递到达的信号。Impala 有两类分组：应用到应用的分组和应用更新器到应用更新器的分组。分组的预定接收方处理分组事件。

③ 发送完成事件：表示一个网络分组已经发送完毕或者发送失败的信号。该事件允许异步网络发送。分组的原始发送方处理发送完成事件。

④ 数据事件：表示感知装置的数据样值已经准备好读取的信号。当前活动应用处理数据事件。

⑤ 装置事件：表示检测到一个装置失效的信号。应用适配器处理装置事件。

当同时发生多个事件时，按顺序处理这些事件。这就排除了不同事件处理器之间同步编程的复杂性。为了防止后续事件处理时延明显增大，要求所有事件处理器在有限时间范围内完成事件的处理。因此，必须将所有阻塞操作（如网络发送）移交给其他系统组件异步执行。

3. 固件对Impala的服务

固件层包含访问和控制各种硬件组件的软件，有以下六个主要固件模块：

① CPU 固件：给 Impala 提供基于系统性能要求的 CPU 方式控制。微型处理器 CPU 可以按照 8 MHz 或者 32 kHz 时钟源运行，32 kHz 时钟源的功耗是 8 MHz 时钟源功耗的一半。当系统执行数据感知和网络通信任务时，Impala 激活高速时钟。Impala 尽可能切换到低速时钟。

② 电台固件：给 Impala 提供数据发送能力和数据接收能力。数据按字节流输入电台和从电台输出。电台固件确保输入电台的字节流被正确封装成物理层分组，以及从不同源节点接收到的物理层分组正确恢复成字节流并作为电台的输出。电台固件完成一个分组接收后就给 Impala 发送一个分组事件。

③ GPS 固件：给 Impala 提供一个获取时间和位置数据的异步接口。首先，GPS 固件配置 GPS 单元，启动感知操作，这个操作耗时 10~60 s，并且根据获取一个位置方位精度要求而变化。同时，GPS 固件分析 GPS 单元的输出信息，识别位置方位。GPS 固件若是获得一个位置方位，则终止感知操作，保存数据，给 Impala 发送一个 GPS 数据事件。

④ Flash 固件：给 Impala 提供 Flash 访问和控制功能。Flash 固件将 Flash 存储器分成 5 节，每节存储不同的信息，比如本地数据、全局数据、诊断信息等。数据连续依次写入每节 Flash，从每节 Flash 连续依次读出数据，可以按 264 B 为一页或者 8 页为一块擦除 Flash 存储器。

⑤ 定时器固件：给 Impala 提供最多高达 8 个软件定时器。任何程序可以声明和释放每个定时器。定时器的所有者可以设置任意定时时间，撤销定时器，复位定时器。分配给应用的定时器一旦定时完毕，定时器固件就立即给 Impala 发送一个应用定时器事件。定时器是 Impala 周期性操作调度的主要机制。定时器固件还维持一个系统时钟，当 CPU 工作时钟为 8 MHz 时系统时钟精确度为 1 ms，系统时钟受到全球 GPS 时间的周期性修正。维持所有 ZebraNet 传感器节点的系统时钟全网同步的能力使得 Impala 网络接口能够使用简单的基于时隙的媒介访问控制机制。

⑥ 看门狗（WatchDog, WDT）固件：给 Impala 提供系统监视和恢复能力。假如系统陷入在错误操作之中或者发生意外故障，那么看门狗固件重新启动系统。

4. Impala对应用的服务

应用层包含 ZebraNet 的全部应用和程序。在 ZebraNet 的第一个版本中，在每个节点上运行的主要应用软件是通信协议软件，用于记录传感器位置数据并将其传递给中心节点。在今后的传感器系统中，应用软件较复杂，包括导航预行计算、传感器数据的过滤与融合软件、数据库查询以及数据通信等软件。

对于数据感知，Impala 将 GPS 有效位置数据存储在 Flash 中。对于数据传输，Impala 执行与其他节点的周期性同步通信。同步数据通信分两个阶段进行。在第一阶段，每个节点采用不可靠广播方式发送一条对等寻找消息，其他节点接收到本条消息就意味着已经找到一个相邻节点，可以将数据转发给这个相邻节点。因此在第二阶段，每个节点采用可靠多目标方式将其位置数据泛洪给所有已找到的单跳相邻节点。为了管理数据存储，每个节点利用本地 Flash 存储节存储本地 GPS 数据，利用全局 Flash 存储节存储其他节点的 GPS 数据。全局 Flash 存储节作为 Impala 存储转发路由的一种媒介。Flash 存储节中原有数据未被擦除，则不能写入新数据，而且最小可擦除单元是一页（264 B），因此按照循环缓存器（依次连续存储数据）方式维护每个 Flash 存储节。

为了支持这些应用活动，Impala 输出三个主要服务。Impala 输出给应用的第一个服务是系统时钟和一个预先分配给应用的定时器，以便执行各种基于定时器的操作（比如周期性同步数据通信）。严格限制应用修改系统时钟或者访问其他系统定时器，否则可能干扰其他已预先安排的系统操作。Impala 输出给应用的第二个服务是受保护的 Flash 读/写操作，以便支持应用数据存储和检索。若应用试图访问未得到授权的 Flash 存储节或者访问不可用 Flash，则拒绝应用访问请求。Impala 输出给应用的第三个服务是一个异步网络传输接口，应用通过该接口将其许多输出消息下传给 Impala 层，并且在传输完成之时会得到 Impala 的通知。

5. 应用编程模型

图 12-3（a）说明了 Impala 基于事件的编程模型。将应用、应用适配器、应用更新器全部编入一个事件处理器集中，当接收到有关事件时事件过滤器就调用这些事件处理器。

应用必须实现四个事件处理器：定时器处理器、分组处理器、发送完成处理器和数据处理器。此外，为了辅助应用适配器查询应用和应用切换，还要求应用实现三个其他程序：应用查询、应用结束、应用初始化。

Impala 用户库包含许多通用编程的应用程序。网络应用程序允许应用将消息封装在分组中异步发送，发送完成后给事件过滤器产生一个发送完成事件。定时器应用程序允许应用根据各种用途设置定时器，比如在某个时刻或者以某个周期时间发送消息时就要设置定时器。一个定时器可以反复地设置、复位、取消。装置应用程序允许应用对装置硬件具有一定程度的控制能力，比如打开、关闭收发信机。应用适配器和应用更新器也可以使用这些编程应用程序。

全局数据结构包括一个统一的感知数据存储映像，以及一个保存应用执行状态的执行结构。①感知数据存储是各个应用均了解的状态资源，一个应用对感知数据存储器的更改会传递给另一个应用。因此，所有应用必须遵从基本存储器结构。但是感知数据存储器的使用细节要满足单个的应用。Impala 定义的统一存储器映像是一张本地感知装置产生的数据

列表、一张从其他节点接收到的数据列表、一本已经成功传递到中心节点的数据记录日志。
②每个应用需要与其事件处理器共享一个执行结构，以便保存网络通信、存储器管理等执行状态。Impala 为所有应用定义了一个简单执行结构，但是其格式是针对特定应用的。

概括起来，图 12-3（b）表示 Impala 基于事件的应用编程模型的一个时序例子，给出了 ZebraNet 中两个通信传感器节点在两个小时数据通信中的事件与事件处理器操作序列。图中箭头表示事情发生时间、文字描述表示每层发生的事情。应用事件处理器响应应用事件，执行路由操作（如对等寻找、数据转发），Impala 事件处理器调用应用程序辅助查询和应用切换。

12.3.3 应用适配器

在 ZebraNet 中，传感器节点装载有多个应用，不同的应用适用于不同的条件，用于将数据传递给中心节点。在两种情况下值得适配：第一种情况是，由于协议的性能、能量效率以及其他属性密切依赖许多因素，所以应用适配器需要处理参数的取值范围、适应其取值的敏感变化；第二种情况是，常常是一个装置对某些协议非常关键而对其他协议却不重要，因此应用适配器应该根据硬件失效情况来做出协议选择，提高强壮性。

采用 Impala 基于事件的编程模型来实现应用适配，当响应事件范围时就需要进行应用适配。有些事件（如定时器事件）表示自最近状态检查以来已通过的时间，然后应用适配器可以选择查询应用或者系统状态，以便决定是否应该执行适配。其他事件（如装置事件）是 Impala 的外部源，表示 Impala 响应的一个外部事件（如某部电台收发信机失效），然后应用适配器应该检查失效收发信机造成的影响，决定是否应该在其周围分派运行另一个应用。

尽管在应用设计中也许还有其他方法可以得到约束性较少或者更加灵活的编程模型，但是因为应用适配器掌握最佳的运行状态信息，所以在 Impala 中由应用适配器做出应用切换决策。应用适配器以单个传感器节点的本地状态为基础，最终应该以全网协调（最适合整个传感器网络）为基础。

1. 适配器功能

定义一个应用参数与系统参数集合来表示运行状态。应用参数由某个特定运行应用掌握的信息组成，系统参数表示 Impala 掌握的信息。

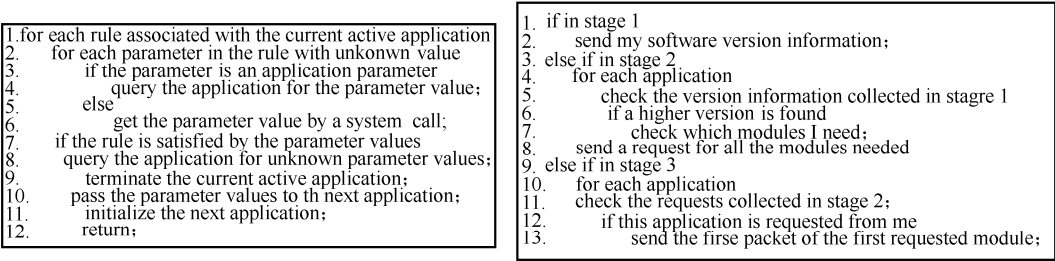
应用参数可能包括以下数据的历史记录、平均值或者总数：直接相邻节点数量；成功转发给对等传感器节点的传感器数据量；存储应用数据的空闲存储器容量等。系统参数包括：电池能量等级；发射机传输距离/发射功率/数据传输速率；节点的地理位置。

每个应用专门跟踪一个应用参数子集，负责报告该应用参数子集的取值。应用适配器有一张应用参数表，记录哪个应用跟踪哪些应用参数。在 Impala 仿真实现中使用 64 个应用参数，所以应用参数表的大小是相当合理的。应用查询和应用切换也使用应用参数表。

为了捕获任何敏感参数的变化，应用适配器周期性查询当前活动应用，以获取其声明跟踪的参数值，提取系统参数值，检查应用切换规则。若满足应用切换规则，则执行应用切换。由于有些应用参数是属性的历史记录，比如最近 k 个周期内直接相邻节点的平均数，所以应用适配器还要将这些参数传递给切换后的下一个应用。

图 12-4（a）给出了应用查询和应用切换的伪码。因为 ZebraNet 中的每个传感器节点每

两个小时中有半个小时处于活动状态，进行网络通信，所以选择在网络活动周期结束时进行应用查询，如图 12-3（b）所示。因此，应用适配器不会干扰应用通信，也不会减少网络带宽。



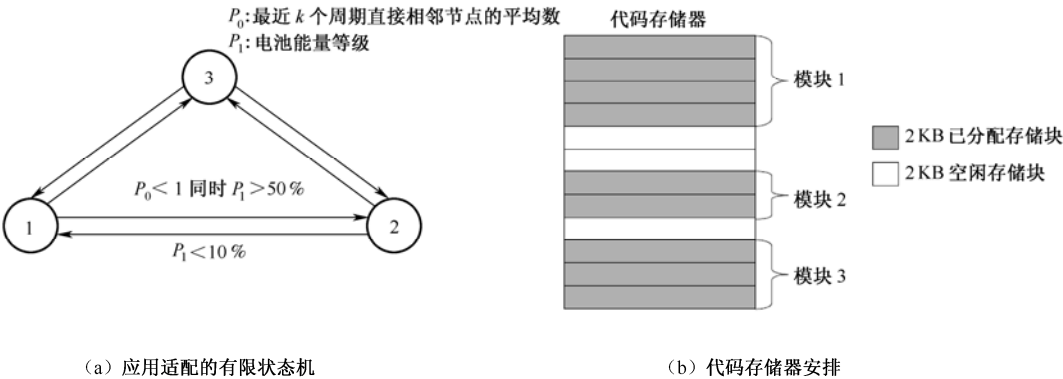
(a) 应用查询和应用切换的伪码

(b) 三阶段按需软件传输策略伪码

图 12-4 Impala 中间件若干伪码

应用适配器完成应用参数值查询后，检查应用适配有限状态机（Adaptation Finite State Machine, AFSM），如图 12-5（a）所示，做出适配决策。AFSM 中的状态符合不同的应用。箭头表示从一个应用到另一个应用的适配转移，箭头旁的参数表达式表示做出应用切换的条件。

例如，假定协议 1 是经过特别选定的协议，使用短距离电台对单个相邻节点发送，而协议 2 是不加选择的泛洪协议，使用远距离电台进行发送。协议 1 能耗较少，产生的网络流量较轻，但是只有在节点相互频繁转发的时候才表现尚好。而当一个传感器孤立在一个远端位置上但是具有足够能量消耗时，采用协议 2 才能够使该节点较为有效地连接其他传感器。因此采用适配规则“假如最近 k 个周期内直接相邻节点的平均数小于 1，并且电池能量在满负荷的 50% 以上，则从协议 1 切换到协议 2”。同理，应该使用另一个切换规则“假如电池能量在满负荷的 10% 以下，则从协议 2 切换到协议 1”。



(a) 应用适配的有限状态机

(b) 代码存储器安排

图 12-5 应用适配的有限状态机、代码存储器安排

2. 基于装置的适配

装置失效的适配方法与参数适配非常相似。有些失效装置会产生装置事件，装置事件能够得到响应。通过周期性查询发现没有响应的硬件部分，从而发现其他失效装置。为了有效

响应装置失效问题，Impala 有一张应用装置表，记录哪些应用依靠哪些硬件装置。在 Impala 原型中，应用装置表的大小允许跟踪 8 个装置，但是在后面的评估中只有三个装置：短距离电台、远距离电台、节点的 GPS 收发信机。应用适配器获悉一个装置失效后，关闭需要该失效装置的协议。假如当前活动应用无效，则切换到一个不需要失效装置的应用。

12.3.4 应用更新器

ZebraNet 具有一些独特特点会改变设计的综合平衡：

- 节点移动性强：在 ZebraNet 中，节点移动性非常强，并且节点按照成群方式运动；
- 网络带宽有限：感知装置频繁收集的数据最终必须全部发送给中心节点，因此网络流量较重，从而导致软件传输可用网络带宽更加有限；
- 更新范围宽：ZebraNet 软件更新范围宽，从小程序缺陷定位到主应用的强化，甚至增加、撤销整个应用。

这些特点意味着应用更新器必须处理以下问题：

- 非完整更新：因为网络带宽窄以及在节点四处移动时可能打断网络连通性，所以节点可以尝试若干次，从网络中收集一个完整更新的要素，因此，非完整更新很普遍；
- 更新与执行：停止软件执行而等待更新完成不切实际，完成更新可能需要一段较长时间，因此，必须一边执行软件，一边同时进行更新；
- 同时更新：更新的网络注入时间可能很长，因此，彼此相近节点发送更新时，节点可能接收到多个非完整更新，其中有些非完整更新可能存在顺序错乱问题；
- 非一致性更新：软件可能变化大，这就意味着模块必须与软件的正确版本号结合在一起；
- 协议传播：应用更新器必须采用有效通信协议，通信协议不仅能够迅速传输软件，而且占用网络带宽最少；
- 代码存储器管理：在 ZebraNet 中，传感器节点装载有四个应用，每个应用包含七个程序模块。传感器网络系统中存在这么多的软件，因此代码存储器管理是面临的一个挑战。

总之，应用更新器的目标是为资源有限的无线移动传感器网络提供有效的软件更新机制。

将 Impala 中动态软件更新的软件管理与传输机制概述如下：对于软件管理，在代码存储器中同时存储完整更新版和非完整更新版。记录完整更新是为了执行，记录非完整更新是为了从最近一次有效更新继续进行更新。对于软件传输，采用按需传输策略。只有在得到请求的条件下，传感器节点才会在交换实际软件代码之前周期性交换软件版本信息。根据对所有传感器节点是否具有最新更新的估计，自动调整软件版本信息交换频率。下面将进一步详细描述。

1. 软件编译、链接、存储器布置

一个程序模块在被注入到网络中之前，需要编译成二进制指令。应用更新器在每个传感器节点上进行链接。在没有接收到同一个更新中的所有模块之前，一个模块不能链接到主程序，不允许模块之间进行对照。这就意味着一个模块的链接与其他模块无关。因此，已经链接的并且在随后版本中还要重复使用的模块不必重新链接。完成链接后，就认为在该传感器节点上“安装”了一个新应用。

代码存储器既存储完整应用，也存储非完整应用。已经接收到一个应用的所有模块，就认为该应用是完整应用。随着时间的流逝，为每个应用保存最高完整版本以及若干个较高非完整版本。为输入模块动态分配存储空间，释放丢弃模块所占用的存储空间。一个模块平均 4 KB，最大 8 KB。每个模块连续占用存储空间，存储空间以 2 KB 为一块依次排列。因此，最小可分配存储块 2 KB，最大申请存储空间是 4 个 2 KB 存储块。图 12-5 (b) 表示代码存储器的安排。

2. 软件版本号

Impala 采用基于模块的版本系统来帮助长期的软件开发和软件更新。每个模块有一个版本号，每个应用有一个版本号。版本号单调递增。

基于模块的版本系统允许选择性地发送软件。节点在交换软件更新前，首先交换应用模块索引，然后只申请传输进行了更改的模块，从而防止传输未更改模块，因此节省网络带宽。

3. 软件传输

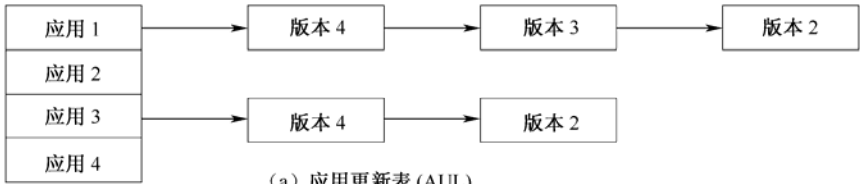
采用按需软件传输策略，分三个阶段完成软件传输，图 12-4 (b) 给出了其伪码。开始时（第一阶段），每个节点将其软件版本信息广播给其他节点，因此每个节点知道其他节点有哪些版本软件。软件版本信息包括完整应用的模块版本号，以及迄今所知的最高版本号。对于安装了较新应用的节点，软件版本信息暗示软件更新的提供节点。对于安装了较旧应用的节点，软件版本信息暗示请求软件更新。因为要求尽可能快地传输软件更新并且仍然保持网络带宽，所以只有在所有节点都没有完整的最新更新时才应该广播软件版本信息。

一个节点接收到其相邻节点广播的软件版本信息后，确定每个应用的有效最高完整版以及哪个节点拥有这个软件版本，然后（向拥有有效最高完整版软件的相邻节点）请求发送自己还没有的模块（第二阶段）。一个已经拥有一些较新版本模块的节点直接请求还没有的较新模块，能够节省不必要的软件传输（因为不会传输已经拥有的那些模块）。在最后一个阶段，具有最高完整版软件的节点发送相邻节点请求的真正软件代码。利用节点的 ID（全网唯一的）做出决断。

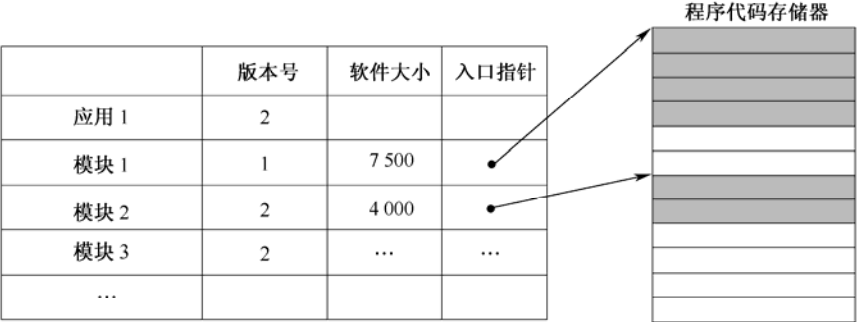
若是每个节点发现其相邻区内并不是每个节点均有迄今所知的最新完整应用，这就说明新应用的节点总数还没有收敛，则在该节点上重复进行上述软件传输的三阶段规程；否则，每个节点发现其相邻区内每个节点均有迄今所知的最新完整应用，则按指数退避重复定时器。重复定时控制在软件开始传输阶段自动将软件传输速率设为最大，而在软件传输快要结束的时候自动减慢软件传输速率。假如新软件的节点总数已经收敛或者新软件还要传播到本区域网络中，那么重复定时控制还能够节省不必要的软件版本信息广播。

4. 软件的接收与安装

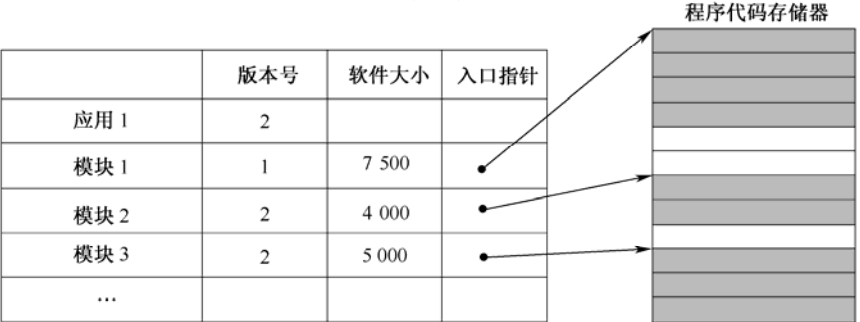
当从网络中接收到一个应用更新时，将其记录在应用更新表（Application Update List, AUL）中，如图 12-6 (a) 所示。AUL 中的每个节点就是一个应用更新记录（Application Update Record, AUR），AUR 的格式如图 12-6 (b) 所示。AUR 的入口指针就是程序模块在代码存储器中的存储位置。对于还没有接收到的程序模块，其入口指针为空指针。按照应用版本号对相同表中的 AUR 进行归类。



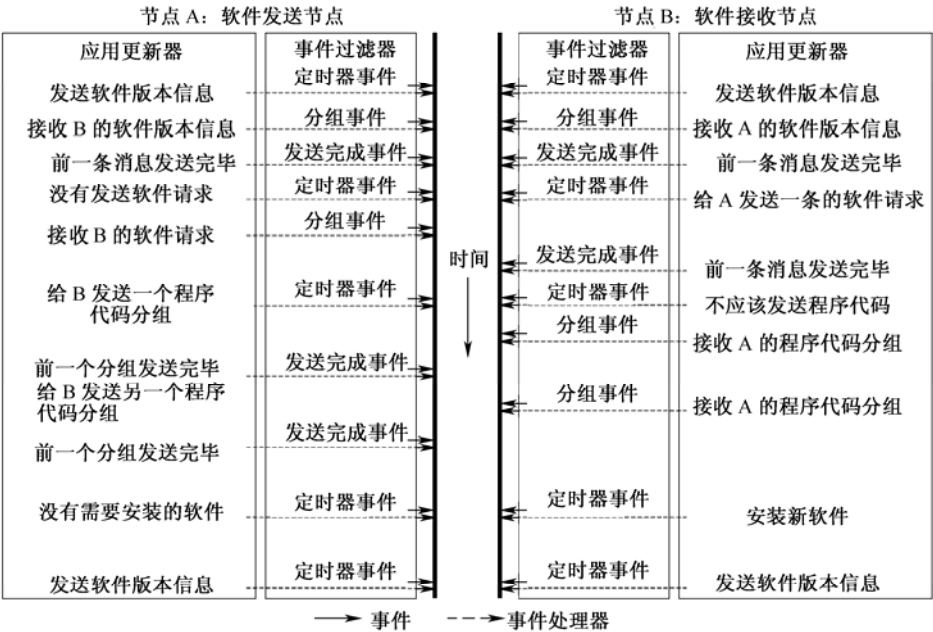
(a) 应用更新表 (AUL)



(b) 应用更新记录 (AUR)



(c) 应用激活表 (AAT)



(d) 基于事件的软件传输、接收、安装

图 12-6 应用更新器

因为接收一个完整更新可能需要很长时间，所以应用更新器尽量保存每个输入更新，以便随时进行软件改进。这样也能节省更新随后表现出来的网络带宽。但是，保存每个非完整更新需要付出极高的存储空间和能耗开销。假如非完整更新过多而造成存储空间不够使用，那么释放一些非完整更新占用的存储空间：从较低版本更新到较高版本更新、从相同应用再到其他应用开始依次释放。

当系统接收到一个更新时，在 AUL 中增加一个 AUR。假如软件接收被中断，那么 AUR 记录丢失哪些模块，下次需要重传这些模块。假如一个模块未被完整接收，则认为该模块丢失。

完成软件接收后，应用更新器就可以在该节点上安装新应用。第一，为了防止意外的编程错误，应用更新器执行简单的安全检查。编程错误包括模块中的无效存储器访问、缺少返回声明，这些编程错误会导致程序执行进入分界线之外。应用更新器还应该检测恶意程序。第二，链接还未链接的模块。最后，应用更新器将 AUR 复制到应用激活表（Application Activation Table, AAT）中。AAT 表条目的格式如图 12-6（c）所示，类似于 AUR，只是入口指针是全部定义的。AAT 作为应用链接表，链接各个应用与主程序。事件过滤器使用 AAT 来调用应用事件处理器，应用适配器使用 AAT 来调用应用程序。AAT 还作为软件版本信息表，在软件传输第一阶段将被广播。AUR 被复制到 AAT 中后，从 AUL 中删除该 AUR 以及较低应用版本号的 AUR。因为 AUR 和 AAT 都使用指向实际程序代码存储单元的存储器指针，所以不需要进行软件的存储器复制。假如当前活动应用正在被更新，那么应用更新器终止旧版本应用，覆盖 AAT 条目，初始化新版本应用。

应用更新器也是采用 Impala 基于事件的编程模型来实现的，如图 12-6（d）所示。图 12-6（d）表示在 ZebraNet 中，在一个两个小时的软件通信周期内两个更新所执行的软件传输、接收、安装。

12.3.5 周期性操作调度

动态地看 Impala 时，Impala 由周期性计算与维护操作（长期感知与通信任务所要求）和随机事件（由传感器网络应用固有的事件驱动属性产生）组成。因此，Impala 系统活动模型有两个方面的独特特性：对于周期性操作，Impala 表现为操作调度器，根据应用目标、硬件约束条件、能量预算安排和协调各种系统操作；对于随机事件，Impala 作为一个事件过滤器，捕获事件并将其分发至不同系统组件，初始化一些列处理。

Impala 利用定时器触发各种操作。图 12-7（a）给出一个重复 ZebraNet 操作的时间序列图：一个节点重复数据的发送/接收操作、获取 GPS 位置的操作、休眠。

Impala 在调度和协调各种系统操作时需要面对许多硬件属性和约束条件。

① 利用 GPS 时间标度能够进行全网操作的时间同步。因为 ZebraNet 传感器节点是在操作中访问全球 GPS 时间，所以传感器节点很容易实现时间同步。这对于所有节点需要同时开/关电台和按分配时隙发送以避免碰撞的网络通信特别重要。

② 电压调整问题阻碍了电台和 GPS 的同时操作。ZebraNet 的电台和 GPS 是两个高电流的组件（见表 12-1）。设计一个允许电台和 GPS 同时操作的电压调整器是一个富有挑战性的问题，而且电压调整器还存在额外功耗。因此，定义网络操作阶段为电台通信阶段、GPS 感知阶段为 GPS 感知阶段，轮流使用电台和 GPS。在 GPS 感知前进行电台通信，这是因为电

台需要同步，GPS 可能需要花费很长时间。

③ 并不小的电台苏醒时间影响网络通信的时间安排。在网络操作阶段开始时刻，所有传感器节点同时打开处于低功率方式下的电台，这至少需要 40 ms，这个时间可能因电台不同而不同。因此，预留一段长度固定的时间给电台苏醒，安排所有节点电台的发送和接收时间，预防电台状态不一致造成的数据丢失。

④ GPS 感知时间长迫使 GPS 感知操作是异步操作。在有些情况下，GPS 只需要 2~6 s 才能够获取一个位置方位，但是获取一个精确位置方位的典型时间是 10~40 s。由于所需时间差异大，所以 GPS 感知任务是一个分割处理任务。首先执行异步感知操作，然后延迟一段时间，接着就是数据交付事件。

⑤ GPS 和 Flash 共享端口，因此禁止同时访问 GPS 和 Flash。微型处理器通常有引脚限制。在 ZebraNet 节点硬件设计中，GPS 和 Flash 共享微处理器的同一个串行接口，因此需要对该端口进行访问方式切换。为此，必须协调 GPS 和 Flash 的操作，防止 GPS 和 Flash 同时进行访问操作。

⑥ 严格的能量预算要求随时节能。能量一直是移动无线传感器网络的关键问题。ZebraNet 节点的电池能量能够支持满额度系统级活动 1~3 天，太阳阵列能够无限期延长活动时间，但是太阳电池区域是有限的，所以需要尽可能节省能量。Impala 采用两种方法节能。第一种方法是，ZebraNet 的 8 min GPS 数据采样周期确保能够捕捉到有效的斑马运动情况，同时冗余数据记录达到最少。这就决定了其他系统活动的频率和数量，因此按照整个系统一直满额度工作的能耗来考虑是不合理的。所以，Impala 有一个休眠阶段，系统在此期间进入低功率方式，对于系统维护只有最少的可用资源。Impala 关闭要进入长时间空闲方式的外设。第二种方法是，尽管 ZebraNet 的能源能够满足典型条件下的能耗需求，但是仍然需要保护系统免受缺乏能量的困扰。因此，Impala 使其操作调度自适应能量可用性。假如能量不足以满足能量需求高阶段（如网络操作阶段、GPS 感知阶段）随后的操作，则快速通过能量需求高阶段。

12.3.6 事件处理模型

Impala 事件处理模型处理三个基本问题。

第一个问题是，传感器网络系统要求高效、基于事件的应用编程接口。事件由硬件中断产生。处理硬件中断不仅涉及相当多的编程，而且还需要详细的硬件知识。因此，Impala 采用一个事件抽象，将各种硬件中断封装成抽象事件，简化应用编程，同时维持应用级处理强度。

Impala 实现四种类型的抽象事件，这四种抽象事件都是 ZebraNet 应用必需的。事件由事件信号源产生和送入队列排队，Impala 事件过滤器完成事件出队列和派发，应用事件处理器完成对事件的处理。图 12-7（b）给出了 Impala 抽象事件和事件处理组件。一个网络分组事件表示一个网络分组已经到达；Impala 网络接口从电台固件接收到一个分组，并验证该分组的有效性后就产生一个网络分组事件。一个网络发送完成事件表示一次网络消息发送已经完成或者失败；Impala 网络接口完成一次网络消息发送或者未能完成一次网络消息发送后就产生一个网络发送完成事件。一个应用定时器事件表示一个预先安排的应用操作的执行时间；应用定时器固件定时结束后就产生一个应用定时器事件。一个 GPS 数据事件表示捕捉到一

个 GPS 位置方位；GPS 固件分析 GPS 单元的输出信息，识别出一个位置方位后就产生一个 GPS 数据事件。

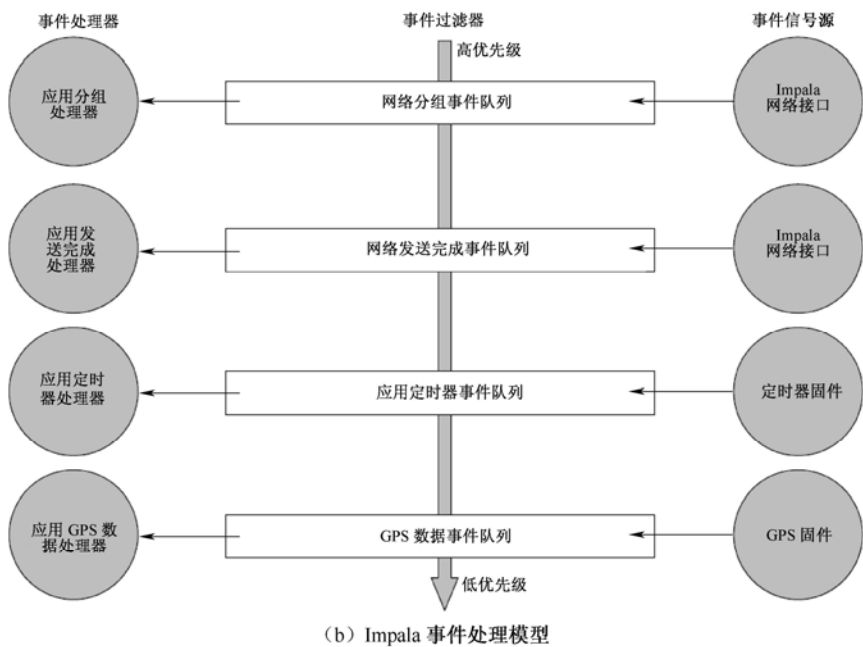
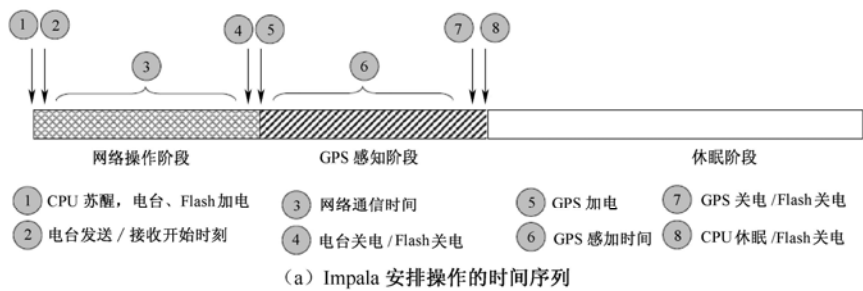


图 12-7 时间序列、事件处理模型

Impala 事件处理模型处理的第二个基本问题是：并发性是传感器网络的固有属性，可能由多个传感器同时捕捉到的信息经过处理后被加载到网络上。此外，有些低级处理具有实时性要求。比如在 ZebraNet 中，电台的字节输出若未得到及时处理就会被丢掉。因此，Impala 采用分层事件处理模型，在短小程序中处理简单的硬件中断，在长而抢先式的程序中处理复杂的软件事件。这不仅达到多个处理流之间的并发性，而且允许与低级处理交替进行，并且必要时可以不顾及高级处理。

为简单起见，在实现 Impala 时采用单线程。硬件中断处理器是不可中断的程序，响应硬件中断，产生软件事件。Impala 事件过滤器是可中断的程序，按照单线程运行，经常检查输入事件，并调用应用事件处理器进行处理。

Impala 事件处理模型处理的第三个基本问题是，在传感器网络中需要对事件进行优先级分配和处理。有些事件紧急，要求立即处理，比如网络分组事件；有些事件有时间限制，但是对稍有延迟不敏感，比如应用定时器事件；其他事件时延容忍范围大，比如 GPS 数据

事件。因此，对事件进行优先级化，使具有不同时间限制的事件按照所要求的顺序得到处理。如图 12-7（b）所示，Impala 事件过滤器为每类事件维护一个队列，每个队列设有相应的事件处理优先级。

12.3.7 Impala网络接口

网络接口作为一个中间件服务，在移动无线传感器网络中是非常重要的。ZebraNet 采用对等通信，但是由于 ZebraNet 弱连通性而选用存储转发路由。为了支持应用层的各种存储转发策略，Impala 网络接口的重点是一个转发跳范围内的网络模型。传感器网络的独特特点对网络模型设计影响甚大。

1. 通信特点对网络会晤的影响

像 ZebraNet 之类的传感器网络应用的特殊通信模式改变着消息模型。数据采集常常是传感器网络的主要任务。特别是在 ZebraNet 中，感知装置频繁收集的数据最终必须全部传递给中心节点。传感器节点常常主动使用泛洪策略，使寻找到到达中心节点的路径的机会最大，因此需要使用多目标传输协议和广播传输协议。此外，有时传输必须是可靠的，但是考虑到能量问题，在重要性较弱时优先使用不可靠传输。Impala 网络服务必须支持这些不同的用法。

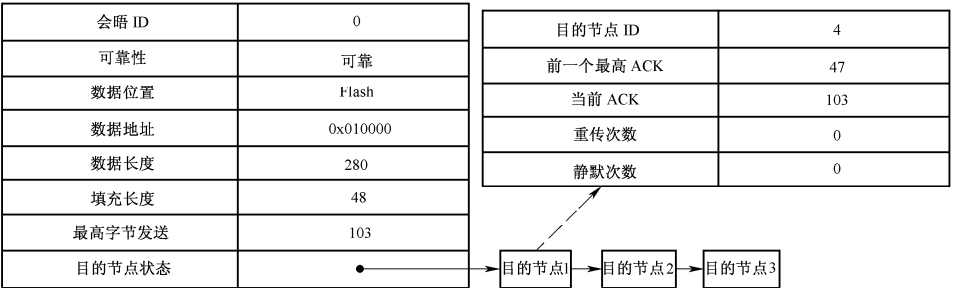
Impala 采用基于会晤的传输控制。一个会晤就是一条由应用指派的消息，具有网络处理语义。会晤大小为 1~32 KB，可以采用单目标、多目标、广播传输方式传输，可以采用可靠传输或者不可靠传输，可以发送 Flash 或者应用 RAM 缓存器中的数据。会晤长度、类型可变支持各种 MAC 层技术。按照无连接方式传输会晤，因为在 ZebraNet 中传感器节点的运动不可预测，因此采用面向连接方式传输会晤极困难。无连接会晤还能减轻计算开销和通信开销。

为了实现会晤，Impala 维护一个发送会晤队列，该队列包含应用已交给 Impala 的所有会晤的会晤描述符。图 12-8（a）给出了发送节点一个会晤描述符包含的信息。每个会晤分得一个 4 bit 的会晤 ID。一个会晤描述符包含一个会晤的属性，并且对于可靠会晤还要跟踪所有目的节点的网络状态。“目的节点状态”连接表是一个已连接记录集合，每个记录保存每个目的节点的分组/ACK 信息。

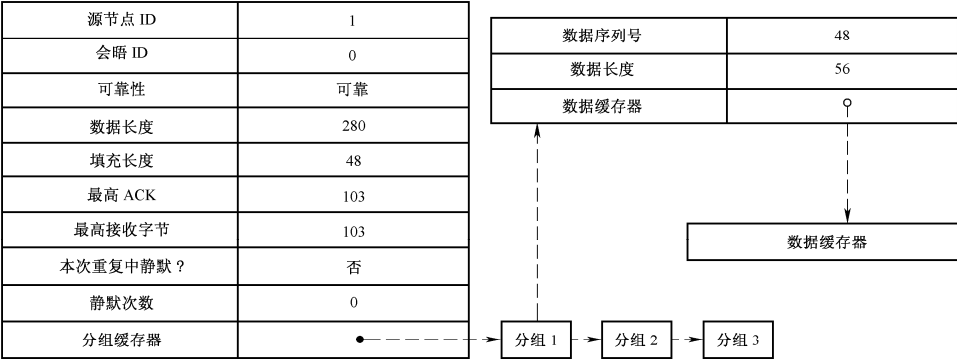
在接收节点一方，Impala 维护一张接收会晤表，该表包含从网络中接收到的所有会晤的会晤拥有节点。图 12-8（b）给出了存储在接收节点的会晤信息。通过源节点 ID 和会晤 ID 能够唯一确定识别每个会晤。为了避免会晤 ID 出现反叠，传感器节点未完成的会晤不能超过 16 个。一个会晤拥有节点包含一个会晤的属性，并且缓存已接收到的但是还未交付给应用的会晤分组。

2. 基于时隙的媒介访问控制

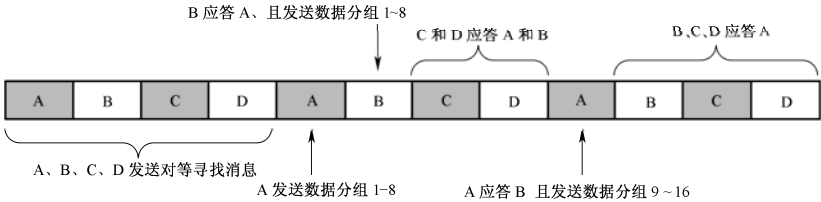
因为 ZebraNet 传感器节点能够访问全网同步的 GPS 时间，所以传感器节点活动易于实现同步。Impala 利用 GPS 时间同步，采用简单、重复、基于时隙的媒介访问控制。作出这种选择的一部分是因为这种媒介访问控制简单（实现代码和能量），但主要还是因为时隙法的重复特性，即 MAC 层总是知道哪些节点应该应答在每个时隙中接收到的分组，因而能够使用简单而高效的超时重传机制。



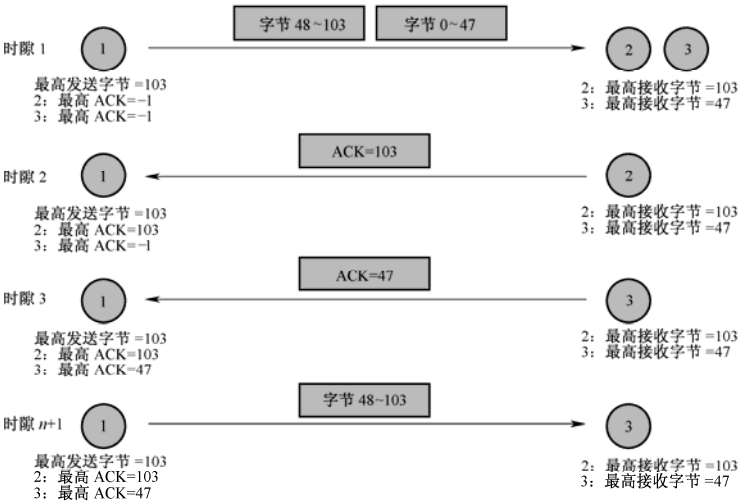
(a) 发送节点的会话描述符



(b) 接收节点对每个会话的分组接收记号



(c) 时隙模型中的数据发送和有关应答



(d) ZebraNet 中的数据发送、应答、重传顺序

图 12-8 Impala 网络接口

在基于时隙的媒介访问控制中，每个传感器节点固定分得一个唯一时隙，并且反复在该时隙上发送。传感器节点在其分得的时隙上发送数据分组和前一个所接收分组的应答。图 12-8 (c) 给出一个例子：在多个传感器节点之间进行时隙化传输。ZebraNet 只有几十个节点，所以这种无可扩展性解决方法是可接受的，且能量效率较高。在大规模网络中，可能需要采用时隙/竞争混合算法，少量节点共享一个时隙。

3. 基于MAC特点的应答优化

由于传感器系统处理能力的限制以及 Flash 存储器复制效率低，所以压缩传统分层协议体系结构，减少数据复制和管理开销。特别是，Impala 统一基于会晤的传输控制与基于时隙的媒介访问控制。

应答、超时重传是可靠会晤传输机制。在每个时隙中，接收节点的 Impala 扫描各个会晤，将待发送的应答封装成一个或者多个分组。发送节点的 Impala 接收到应答分组后，提取应答，对其每个会晤的 ACK 记录作相应更新。

基于会晤的传输控制能够意识到基于时隙的 MAC 协议，采用简单而高效的超时/重传机制。每个传感器节点分得一个发送时隙，到达该时隙时刻就进入发送，因而知道在前一轮发送的分组应该已经被目的节点所接收和应答。因此，假如这些发送分组未被应答，那么该节点就知道要么会晤要么应答已经丢失，因此重传未被应答的分组。这种超时重传机制能够在最早时刻进行重传，提高了通信性能。图 12-8 (d) 表示一个发送节点和两个接收节点之间跨越几个时隙的发送-应答-重传规程。

超时机制在移动无线传感器网络中非常重要，这是因为节点可能移动而离开所在的通信范围。Impala 对同一个分组集合重传四次后或者通过两次重传目的节点就已经静默，则会发生针对该目的节点的超时。假如所有目的节点已经超时，那么发送节点的 Impala 终止会晤，报告一个会晤失败。同样地，源节点在其连续重复的 4 个发送时隙中均一直处于静默后，接收节点的 Impala 就会发生一个会晤超时。

4. 数据缓存约束条件

传统网络模型要求足够存储器缓存数据。但是传感器网络有限的存储器必然要求重新考虑数据缓存模型。ZebraNet 传感器节点有足够的 Flash 存储器，但是只有 2 KB RAM。因为 Flash 速度慢，且对 Flash 重写有限制，所以 Impala 采用如下两个机制来减少数据缓存所需要的存储器容量。

第一个机制是使用数据检索替代数据缓存。无论正在发送的数据是来自 Flash 存储器，还是来自 RAM 应用缓存器，Impala 均保持数据存储位置固定不变，记录其存储单元，直接从其存储单元读出并发送，而不是将其复制到一个大网络缓存器中。数据检索使网络传输需要的 RAM 容量最少，并且能够发送大批量数据，因此网络吞吐量不会被有限存储器所抑制。

第二个机制是在网络接收过程中使用数据缓存替代会晤缓存。Impala 不用缓存一个可能很大的会晤后才将其交付给应用，而是按次序缓存单个分组。即使随后同一个会晤的分组处在传输过程之中，Impala 也仍然立即将所接收分组交付给应用。这种渐进式交付也非常适用于很多感知应用面向流的特性。

5. 分组与分组事件交付

分组交付不同于会晤交付，影响应用编程风格，但是分组交付适用于渐进式、基于流的处理。在面向 TCP 的应用编程中，应用具有足够的 TCP 缓存器，能够完成处理一个发送节点的一个会晤后再切换到另一个发送节点的另一个会晤的处理。这种编程风格不适合 ZebraNet。应用不是处理语义应该完整的会晤数据并将其立即存储在 Flash 存储器中，而是必须处理不同发送节点交错的不完整分组数据。

在 ZebraNet 中，电台硬件的短小物理分组也影响着分组格式的设计。因为原始分组短小，所以传统的多级网络协议分组头是一个相当大的通信开销。因此 Impala 采用的分组格式包含一个微小分组头。微小分组头是前面已发送的完整长度分组头（每个会晤只发送一次，适用于该会晤的所有分组）的缩减表示。

接收节点的 Impala 接收到一个会晤分组后，检查其分组头是否与现有某个会晤拥有节点关联；假如与现有某个会晤拥有节点关联，那么 Impala 接着检查该分组序列号，按序缓存分组，丢掉乱序分组和重复分组。假如该分组头不关联现有会晤拥有节点，那么 Impala 检查该分组是否包含一个会晤头；假如包含一个会晤头，那么 Impala 创建一个新的会晤拥有节点并缓存这个分组；假如不包含会晤头，那么 Impala 丢掉该分组。

6. 异步网络传输

如前所述，Impala 有一个基于事件的应用编程模型，实质上是通过创建一个包含甚广的事件处理器全集来对应用进行编程的。要求所有应用级事件处理器在有限时间内完成事件的处理，其原因有两点：第一，有了这个要求就可以对有关 Impala 对外部事件的最大响应时间做出保证；第二，这个要求有助于防止出现表现甚差的应用活锁节点。因为应用不能对长时间事件（如网络传输）等待任意长时间，因此将这种事件移交给 Impala 异步处理。Impala 给应用提供异步传输模型，该模型包含数据发送和数据发送已经完成信息的网络钩。

12.3.8 Impala 评估

为了评估 Impala 在实际传感器网络中的开销，在移动装置 HP/Compaq iPAQ 袖珍 PC 上实现 Impala 原型，重点测量 Impala 在事件交付和事件处理过程中的开销。然后在 ZebraNet 硬件节点上实现 Impala 中间件，包括固件层、Impala 层、基准应用层，针对以下系统方面评估 Impala：存储器的固定需要和动态需求，操作调度与事件处理的开销，网络接口性能。

1. 实现详情

用来实现 Impala 原型的 iPAQ 袖珍 PC 具有如下配置：206 MHz CPU 运行 Linux5.3，具有 XipaqGUI，32 MB 闪存 RAM 作为主存储器，16 MB 闪存 ROM 作为文件系统。采用一对袖珍 PC 进行实验。对于 Impala 中间件层，实现应用适配器、应用更新器、事件过滤器、网络应用程序、定时器应用程序。对于应用层，实现两个应用协议：经过特选的、基于历史（节点对中心节点数据交付历史纪录）的协议（以下简称历史协议），不加选择的泛洪协议。表 12-2 是应用程序的存储器概况。

在两台运行 Impala 的 iPAQ 节点上进行实验，实验时间为 3 个 1 min 周期。每个实验周期（1 min）等效于 ZebaraNet 一个两小时的数据/软件通信周期。两个 iPAQ 节点均装载有两个协议。两个节点的泛洪协议的版本号相同，但是对于历史协议，节点 0 的分组处理器版本号比节点 1 新。两个节点从历史协议开始运行。在第一个周期中，较新的分组处理器从节点 0 发送给节点 1，并安装在节点 1 上。但是根据历史协议，任何一个节点不会认为另一个节点是合适的发送目标，所以不会发送传感器数据。在第二个周期，由于两个节点在前一个周期找到的相邻节点太少（在这种条件下历史协议表现不好），所以两个节点均切换到泛洪协议，从而发生数据交换。应用更新器知道本次应用版本是一致的，因此不会进行软件传输，应用更新器的定时器也退避 1 个周期。在最后一个周期，由于两个节点已经完成了足够的数据分发，所以两个节点又切换到历史协议，本次节点 1 运行较新版的历史协议。应用更新器处于空闲状态。

表 12-2 应用程序的存储器概况（iPAQ 节点）

历史协议的程序模块	指令大小（B）	泛洪协议的程序模块	指令大小（B）
数据处理器	1 104	数据处理器	1 044
定时器处理器	2 228	定时器处理器	1 752
分组处理器	808	分组处理器	560
发送完成处理器	1 360	发送完成处理器	1 360
应用初始化	1 140	应用初始化	568
应用结束	28	应用结束	28
应用查询	360	应用查询	108

2. 开销测试结果

Impala 在检查完事件的目的地后才交付事件，从而耽搁事件的处理，而事件延迟处理又会引起事件交付时延，因为分组事件、发送完成事件、定时器事件只涉及一个分支的比较指令。数据事件只由当前活动应用来处理，不需要检查目的地，因此没有时延。

Impala 处理发送给自己的事件。有些事件处理时间是 Impala 固有的（集成法没有），比如应用查询和应用切换时间。其他事件处理时间是 Impala、集成法共有的，比如软件传输、接收、安装的时间。表 12-3 列出了 Impala 事件和应用事件的处理时间的相对值。一般地，Impala 事件的发生频率低于应用事件。在表 12-3 中，应用更新器接收和处理一个程序代码分组的时间比其他时间大许多，这是因为在 Impala 原型中，将所收程序代码分组写入到 iPAQ 闪存 ROM 的文件中，这种文件写操作非常缓慢。事实上，文件写操作占总时间的 95%。

表 12-3 Impala 事件处理时间（iPAQ 节点）

处 理 类 型	处 理 详 情	处理时间（μs）
应用执行	对等寻找消息指令和传输	3 211
	对等寻找消息接收和处理	14
	180 B 数据分组打包和传输	3 563
	180 B 数据分组接收和处理	38
应用适配	应用查询和应用切换	1 301

续表

处 理 类 型	处 理 详 情	处理时间 (μs)
应用更新	软件版本信息打包和传输	478
	软件版本信息接收和处理	102
	软件请求打包和传输	1 230
	软件请求接收和处理	58
	256 B 程序代码分组打包和传输	2 024
	256 B 程序代码分组接收和处理	12 617
	软件安装	2 329

3. 网络编程的效率

Impala 的一个重要设计目标就是实现最佳软件传输速率，同时占用最少网络带宽。为了证实 Impala 按需软件传输策略的效率，做仿真实验，对整个传感器网络进行重新编程。在实验中，50 个传感器节点在 40 km×40 km 区域内四处移动，中心节点给其邻区中每个节点广播软件更新，然后每个相邻节点采用对等通信将软件更新传播给更远的节点。节点无线传输距离为 4 000 m。节点上的应用更新器每隔 2 个小时苏醒一次，交换软件更新。评估两个更新策略。第一个更新策略是每次应用更新器苏醒时以一定概率将软件更新广播给其他节点。已经接收到最新软件更新的节点丢掉重复接收的软件更新。第二个更新策略就是 Impala 的按需软件传输策略。

图 12-9 (a) 表示一个软件更新注入到网络所需要的时间。定时器 0 表示网络中中心节点首先到达软件更新的时间点。当概率等于 1 时，概率广播策略达到最大传输速率，这是理想情况。图 12-9 (a) 说明采用 Impala 按需软件传输策略，网络注入速率在开始时迅速提高，而在快要结束时下降，非常接近理想曲线，优于采用较低概率时的概率广播曲线。但是 Impala 按需软件传输策略会丢掉立即可得的更新机会。例如，当一个已更新节点群中的一个节点遇到一个未更新节点群中的一个节点时，两个节点均认为更新规程已经收敛，退避各自的软件版本信息广播时间定时器（由于两个节点已经知道更新的同类节点总数）。在这种情况下，在退避时间结束之前，两个节点不能相互寻找到对方，从而延迟了软件更新。

对于 Impala 按需软件传输策略，软件传输量大约随着已更新节点数的增大而递增。对于概率广播，软件传输量递增更快，并且随着时间的流逝而线性递增。两种策略占用的实际网络带宽的差异可能更大，这是因为 Impala 按需软件传输策略只选择已更新模块发送，而概率广播策略不区分模块是否已被更新而发送所有的模块。但是，Impala 按需软件传输策略存在通信开销，即软件版本信息和软件请求消息的通信开销。完成网络重新编程时，总共发送了 247 条软件版本信息消息、47 条软件请求消息，而真正的软件传输 36 个。

4. 适配的优点

Impala 的另一个重要设计目标是实现各个协议的“智能”安排以适应各种条件，以及最大程度地提高整个系统的性能。为了证实 Impala 协议适配的功效，做几个仿真实验，重点观测 Impala 对系统三个方面的改进：仿真方法、路由性能、能量效率。

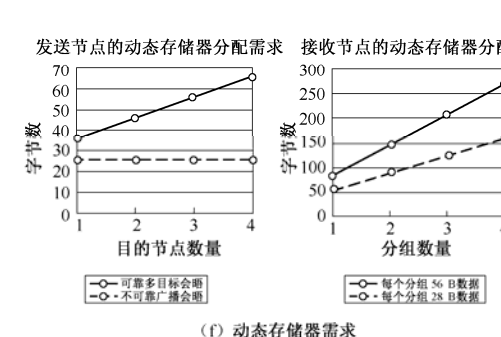
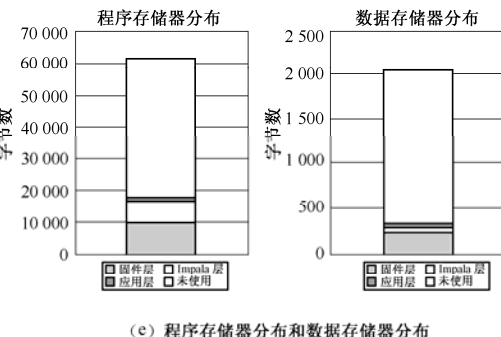
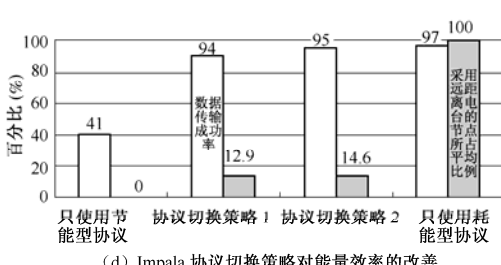
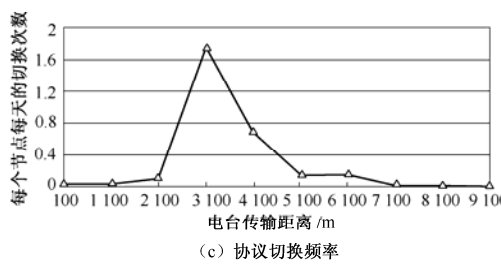
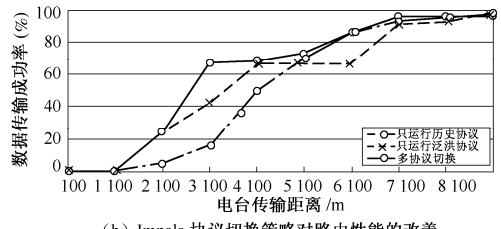
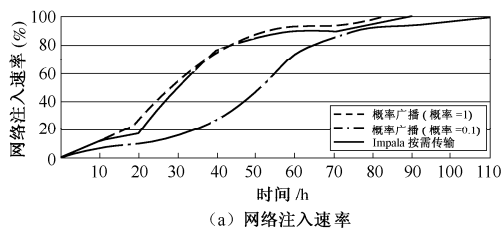


图 12-9 Impala 中间件的评估

(1) 仿真方法

利用传感器网络仿真器 ZnetSim 的升级版^[10]进行仿真实验。50 个传感器节点在 40 km×40 km 区域内四处移动。每个传感器节点有两部电台：一部短距离电台、一部远距离电台。在网络中对若干个路由协议进行编程。直接传输协议作为基准协议，不涉及任何对等转发。每个节点若是处在中心节点的覆盖范围内，则只转发数据给中心节点。泛洪协议将数据发送给每个相邻节点。历史协议根据每个节点曾经成功转发数据给中心节点的历史记录而假定该节点有一个分层等级，对一个节点处在中心节点覆盖范围内的概率进行编码，然后将数据发送给分层等级最高的那个相邻节点。每个仿真实验仿真传感器在 30 天内的活动，计算路由成功率（即最终被中心节点所接收到的数据所占比率）。

(2) 路由性能的提高

第一个仿真实验验证协议安排能够根据不同网络连通性调整传感器通信模式、实现最佳路由成功率。在本次实验中，每个传感器节点可以安排两个协议：泛洪协议和历史协议，两个协议使用同一个电台装置，电台传输距离 100~9 100 m。图 12-9 (b) 中的两条虚线表示基本情况：所有传感器节点总是运行相同的协议。当电台传输距离短时，网络连通性差，传感器几乎找不到相邻节点；此时，泛洪协议优于历史协议，这是因为将数据发送给每个相邻节点，找到到达中心节点的数据路径的机会最大。但是，当电台传输距离长时，网络连通性

好, 传感器可以只在其他传感器不在附近时才发送; 因此, 此时泛洪协议优于历史协议, 这是因为对网络注入过多数据就会受到有限网络带宽的限制。因此, 采用下列协议切换策略使每个传感器适应网络连通性和带宽可用性:

- 开始时运行历史协议;
- 假如在运行历史协议, 并且在最近半小时协议通信窗口中遇到的直接相邻节点平均数小于 16 个以及非直接相邻节点平均数小于 42 个, 则切换到泛洪协议;
- 假如在运行泛洪协议, 并且直接相邻节点平均数大于 20 个或者非直接相邻节点的平均数大于 42 个, 则切换到历史协议;
- 否则, 继续运行原协议。

图 12-9 (b) 中的实线是上述切换策略的实验结果。对于传输距离极短或者很远的电台, 总是遵循两条非切换曲线中较好的一条。对于中等传输距离的电台, 切换曲线总是优于两条非切换曲线。这个实验结果进一步证实: 当网络连通性比较复杂 (即有些节点成群聚集在一起, 而其他节点相距甚远) 时, Impala 协议安排能够根据本地特定网络条件选择合适的路由协议。

图 12-9 (c) 表示本实验中协议切换的频率。在 3 100 m 处, 协议切换最频繁, 达到最大程度的性能改善, 每个传感器节点的平均切换开销小于每天两次。

(3) 能量效率的提高

在仿真实验中, 每个传感器节点安排两个协议, 以提高能量效率。这两个协议是: 采用短传输距离 (1 000 m) 电台发送的历史协议, 采用远传输距离 (5 000 m) 电台发送的泛洪协议。

历史协议不仅有低功率电台, 而且在数据发送时是保守的, 因为传感器只是将数据发送给对中心节点数据交付历史纪录优于自己的节点。泛洪协议采用大功率电台主动发送数据, 路由性能强壮, 但是能量效率低。显然, 这两个协议具有互补性: 考虑到能量效率时应该使用历史协议, 但是当数据交付成功率下降至过低时应该使用泛洪协议 (作为一个补充协议来使用)。

对于每个传感器节点, 采用如下协议切换策略:

- 开始时运行历史协议;
- 假如在运行历史协议并且在最近 8 h 内遇到的直接相邻节点平均数小于 m 个 (m 是一个可调参数) 或者本节点在 12 h 以上没有对任何一个对等传感器节点 (或者中心节点) 进行过发送, 则切换到泛洪协议;
- 假如在运行泛洪协议并且本节点至少对一个对等传感器节点 (或者中心节点) 进行过发送, 则切换到历史协议;
- 否则, 继续运行原协议。

图 12-9 (d) 表示上述协议切换策略的实验结果。对于第一个策略, 设 $m=1$; 对于第二个策略, 设 $m=4$ 。从图 12-9 (d) 中看到: 尽管耗能型协议的使用时间只占总时间的 15%, 但是采用 Impala 协议切换策略显然能够将节能型协议路由性能提高至与耗能型协议路由性能大致相当。需要注意的是: 本实验不同于前一个仿真实验, 这里假定网络带宽不受限制, 也不是协议的一个约束条件。

5. 固定存储器分布

前面介绍的是 Impala 仿真结果, 从本节开始通过在 ZebraNet 硬件节点 (见图 12-2) 上

实现 Impala 中间件来评估 Impala。因为 ZebraNet 系统的存储器容量限制很苛刻，所以要求 Impala 代码量最少且使用的 RAM 也最少。图 12-9 (e) 表示不同系统层的程序存储器分布和数据存储器分布。

对于程序存储器，网络接口需要 5 712 B 指令，是 Impala 层需要存储单元最多的一个组件。Flash、GPS、定时器模块是固件层的主要组件。应用层只实现了一个基本应用，所以所需存储单元最少。

对于数据存储器，Impala 层网络接口声明 51 B 数据存储器。在固件层，GPS 模块需要一个 125 B 的缓存器用于接收 GPS 单元的信息，电台模块需要一个 64 B 的缓存器用于接收电台的分组。

如图 12-9 (e) 所示，程序代码量（当前版）不足总程序存储器容量的 1/3，静态分配的 RAM 不足总 RAM 的 1/6，因此留有足够 RAM 供动态分配使用和今后使用。

6. 动态存储器需求

Impala 网络接口为空中网络发送和接收声明和释放缓存器，因此需要动态存储器分配。

在发送节点一方，Impala 为应用会晤分配缓存器，以便于应用会晤异步传输。可靠多目标会晤的缓存器容量线性正比于目的节点数量，而不可靠广播会晤的缓存器容量恒定不变。图 12-9 (f) 给出了最多缓存 4 个目的节点的会晤的动态存储器容量要求。因为 Flash 中的数据是由网络接口来检索而不是缓存的，所以发送节点方的存储器容量需求得到大幅度下降。

在接收节点一方，Impala 分配缓存器，用于缓存还没有交付给应用的分组。缓存器容量线性正比于所缓存的分组数量。图 12-9 (f) 给出了最多缓存 4 个分组的动态存储器容量要求。

7. 操作调度与事件处理开销

操作调度开销常常源于装置工作前后的控制、后续操作的定时器设置。操作调度开销最低化对于 ZebraNet 系统的合理实现非常关键。假如操作安排占用过多时间，那么操作本身可能被延迟。这可能产生意外结果，比如引起各个节点丢失时间同步。

表 12-4 给出了安排各种 Impala 操作所需要的 CPU 时间。打开电台需要 50 ms，这是因为 Impala 必须等待电台从低功率方式下苏醒，因此在所有电台开始苏醒时预留一段电台苏醒时间。关闭电台需要 11 ms，这是因为 Impala 必须等待表示电台发送缓存器空的引脚信号。所有其他操作调度开销小于 1 ms。

Impala 将事件交付给应用的时延决定响应事件的快慢。表 12-5 给出所有应用事件的事件处理时间开销。比如以表 12-5 中的网络分组事件为例，其处理时间开销涉及事件在队列中的进出、调用上层事件处理器、释放动态分配的存储缓存器。

表 12-4 操作调度开销

调度类型	调度时刻	Impala 活动	时 间
CPU 调度	系统活动时间开始时刻	使 CPU 按照高速时钟工作	3 127 时钟周期
	系统休眠时间开始时刻	使 CPU 按照低速时钟工作	38 时钟周期

续表

调度类型	调度时刻	Impala 活动	时间
电台和 Flash 调度	网络时间开始时刻	设置第一个发送时隙，打开电台，唤醒 Flash	50 ms
	最后一个时隙之外的所有时隙	设置下一个发送时隙	260 时钟周期
	最后一个时隙	设置网络清除时间、电台和 Flash 关闭时间	265 时钟周期
	网络时间结束时刻	清除未完成的网络会晤，关掉电台和 Flash 电源，设置下一个网络时间	11 ms
GPS 调度	GPS 感知时间开始时刻	初始化 GPS 感知，设置 GPS 感知结束时间	1 247 时钟周期
	GPS 感知时间结束时刻	格式化 GPS 数据，关掉 GPS 电源，发送一个 GPS 数据事件，设置下一个 GPS 感知时间	2 550 时钟周期

表 12-5 事件处理开销

Impala 事件处理活动	时间（时钟周期）
交付和移动一个网络分组事件	88
交付和移动一个应用定时器事件	16
交付和移动一个 GPS 数据事件	17

12.4 传感器信息网络化体系结构（SINA）

传感器信息网络化体系结构（Sensor Information Networking Architecture，SINA）将一个传感器网络模拟为一个大型分布式目标群，起着中间件的作用，允许传感器应用向网内发送查询和控制任务、从网内收集应答和结果、监视网内变化，如图 12-10（a）所示。每个传感器节点运行 SINA 模块。SINA 模块提供传感器信息自适应结构，辅助查询、事件监视、任务分配能力，如图 12-10（b）所示。

在通常的分布式数据库中，信息分布在若干个地点，而传感器网络中的信息分布地点数量等于传感器数量，每个传感器收集的信息自然成为该节点的一个组成部分（或者属性）。为了支持能量高效以及可扩展操作，传感器节点自动分群。由于传感器信息的数据中心特性，所以采用基于属性的命名方法（而不是采用直接地址）能够更加有效地访问传感器信息。

12.4.1 SINA的功能组成

SINA 中间件由分层分群、基于属性的命名、位置意识功能块组成。下面分别加以详述。

1. 分层分群

为了便于传感器网络内部的可扩展操作，应该根据传感器节点的能量等级以及邻近关系对传感器节点进行分群组织，如图 12-10（c）所示。递归应用分群方法建立分层分群网络结构，如图 12-10（d）所示。在一个分群内部选择一个群首节点负责执行信息过滤、融合，比如周期性计算本分群覆盖范围内的平均温度。当群首失效或者电池能量较低时，应该重新初始化分群过程。当分群分层结构不合适时，应用将传感器网络看做只有一层的分群结构，此

时每个节点本身就是一个群首。可以采用第 6 章介绍的分群协议（LEACH、TTDD、HEED）、也可以使用其他分群协议构建分群分层网络结构。

2. 基于属性的命名

当网络规模很大时，关注单个传感器节点不切实际。用户更应该关注查询哪个（哪些）区域的温度高于 100°F，或者东南区的平均温度，而不是关心特定传感器 ID=101 的温度。为了推动传感器查询的数据中心特性，优选采用基于属性的命名方法。例如，名称[type=temperature, location=N-E, temperature=103]描述东北区所有传感器感知的温度为 103°F，这些传感器应答的查询是“那个（那些）区域的温度高于 100°F”。

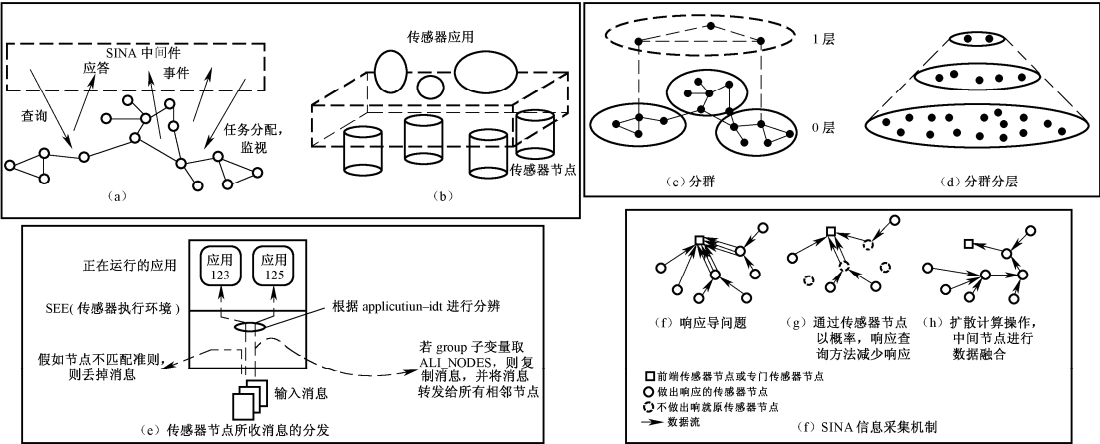


图 12-10 SINA 说明图

3. 位置意识

传感器节点在自然环境中工作，掌握自己的物理空间位置是非常关键的，可以采用多种方法获取传感器节点的位置信息，比如可以采用第 7 章介绍的网络定位技术（如 MDS、TPS、噪声距离测量法），也可以采用其他网络定位技术。在条件允许的情况下也可以采取 GPS。GPS 提供绝对位置信息，但是考虑到经济原因，只给一个传感器节点子集配备 GPS 接收机，这些节点周期性发送信标信令，将自己的位置信息告诉其他节点，起着位置参考的作用，其他没有配备 GPS 接收机的节点就能够大致确定自己的地理位置。

将上述三个功能组成综合起来，就能够进行如下两种查询：

① “哪个（哪些）区域的温度高于 100°F？”理论上，广播这个查询，每个网络节点对这个查询做出应答。尽管可能得到最佳应答结果，但是查询会遇到响应时间长的的问题。在实际中，每个群首可以周期性更新其成员节点的感知温度，按照多目标传输方式将这个查询发送给群首，并且只由群首做出应答。这样能够改善查询响应时间，但是代价是应答精确度有所下降。在严格定时限制下，可以只由高层群首应答查询。

② “东南区的平均温度是多少？”每个群首周期性更新和存储其平均温度。应该将这个查询只发送给东南区的传感器节点。

12.4.2 信息抽象

在 SINA 中，将一个传感器网络看成一个数据表集合，每张数据表包含每个传感器节点的一个属性集合，每个属性称为一个蜂窝，传感器网络的数据表集合表示联合电子数据表（Associative Spreadsheet），通过基于属性的名称表示联合电子数据表中的各个蜂窝。开始时，每个传感器节点的数据表只有少量预先定义的属性。这些传感器节点一旦布置完毕并且形成一个传感器网络的时候，就可以通过评估有效蜂窝结构表达式，可以从其他蜂窝获取信息，从而接收其他节点的请求（比如来自其群首的请求）而建立新蜂窝，调用系统定义的函数，或者累积来自其他数据表的信息。

每个新建蜂窝必须得到唯一命名并且成为某个节点的属性，该属性或者取单值（如剩余电池能量）或者取多个数值（比如过去 30 min 内温度记录的变化）。通过综合运用分层分群机制和基于属性的命名法，SINA 提供一个功能强大的操作集，处理传感器节点间的数据访问和数据累积。SINA 采用联合电子数据表机制更易于实现节点间通过属性命名的相互交互。

12.4.3 传感器查询与任务分配语言（SCTL）

传感器查询与任务分配语言（Sensor Query and Tasking Language, SCTL）是 SINA 的一个组成部分，起着传感器应用与 SINA 中间件之间的编程接口作用。SCTL 是一种程序脚本语言，具有面向对象的特点，灵活、紧凑，能够解释简单的公开查询声明。SCTL 语言结构包括：算数操作符（+，-，*，/），比较操作符（==，!=，<，>），布尔操作符（AND，OR，NOT），分配（assignment），有条件结构（if-then-else），循环结构（while），目标实例化（new），事件处理结构（upon）。没有变量声明块，因此可以按需创建任何类型变量。SCTL 大多数语言结构的使用方法与其他程序语言相同。

表 12-6 SCTL 封装头常用组成域

变 量		含 义
sender		一个 SCTL 消息封装头的发送节点
	receiver	由两个子变量 group、criteria 说明的可能接收节点
	group	说明接收节点组的接收节点子变量，其取值只能是 ALL_NODES 或者 NEIGHBOES
	criteria	说明接收节点选择准则的接收节点子变量
application-id		相同传感器网络中每个应用的唯一 ID 号
num-hop		离网关节点（中心节点）的转发跳数距离
language		说明有效载荷域（content）中采用的语言
content		有效载荷，包括一个程序、一条消息或者返回值
	with(optional)	程序中使用的可选参数数组，从发送节点传递给接收节点
	parameter	可重复的子变量，含变量 type、name、value
	type	该参数的数据类型
	name	该参数的名称
	value	该参数的取值

SQTL 提供传感器硬件访问原语[比如 `getTemperatureSensor()`、`turnOn()`、`turOff()`]、位置意识原语[比如 `isNorthOf()`、`isNear()`、`isNeighbor()`]以及通信原语[比如 `tell()`、`execute()`、`send()`]，提供实现基本数据结构的类（比如排列、链表）。数据累积函数（比如最大、最小、平均）也适用于数据结构。SQTL 还提供一个事件处理结构，该结构适用于这类传感器网络应用：传感器节点被编程为处理异步事件（比如接收一条消息或者定时器触发的一个事件）。程序员通过使用 `upon` 结构就能够创建一个事件处理模块。目前，SQTL 支持以下三类事件：

- 一个传感器节点接收到一条消息时产生的事件，采用关键字 `receive` 定义；
- 定时器周期性触发的事件，采用关键字 `every` 定义；
- 定时器定时结束时产生的事件，采用关键字 `expire` 定义。

一条 SQTL 消息包含一个脚本，网络中任何一个传感器节点解释和执行本条 SQTL 消息。为了使一个脚本针对一个特定接收节点或者一个特定接收节点组，采用 SQTL 封装头封装 SQTL 消息。SQTL 封装头作为消息头，说明发送节点、接收节点、这些接收节点上运行的特定应用以及该应用的参数。

SQTL 封装头采用 XML 语法定义应用层头，应用头说明属性名称的复杂寻址方法。表 12-6 概括了 SQTL 封装头常用组成域。

12.4.4 传感器执行环境（SEE）

在每个传感器节点上运行的传感器执行环境（Sensor Execution Environment，SEE）负责分发输入消息、检查到达的所有 SQTL 消息、对 SQTL 消息中说明的每种动作采取适当操作。SEE 检查 SQTL 消息内的 `receiver` 变量，并根据其取值决定是否将该条 SQTL 消息转发给下一个转发跳。其 `group` 子变量取值为 `ALL_NODES` 的消息将被转播给网络中的每个传感器节点，其 `group` 子变量取值为 `NEIGHBORS` 的消息只被转发给本节点的一跳相邻节点。比较 `criteria` 域中说明的属性-取值对表中的属性名称、接收节点数据表中存储的接收节点属性：假如该节点的属性满足准则，则 SEE 接收该消息。

一旦一个 SQTL 脚本从前端节点（一个与网络直接连接的特殊节点）注入到一个或者多个传感器节点，那么这个脚本就自动向前传递给其他节点，以便完成所分配的任务。然后在每个传感器节点产生一个结果后产生一条 `tell` 消息，并将该消息回送给请求节点，请求节点通常就是将脚本发送下来的上行节点。图 12-10（e）描述了 SEE 的输入消息分发过程。

SEE 除了完成输入 SQTL 消息的分解之外，还要处理所有正在运行中的应用输出的 SQTL 消息：采用低层通信机制将应用输出 SQTL 消息分发给本消息头中 `receiver` 变量指定的目标节点。SEE 可能将属性名称转换成唯一的链路层可用数字地址。否则，采用链路层广播该消息。

12.4.5 信息收集方法

对于利用 SINA 体系结构的应用，传感器节点间的低层通信机制起着重要作用。通过提供支持特定应用要求的高效数据分发和信息采集，SINA 抽象化低层通信，使高层传感器应用远离低层通信。当用户提交查询时，并不要求用户明确定义如何从传感器网络内收集信息。SINA 体系结构根据查询类型和当前网络状态选择最合适的数据分发和信息收集方法。前端节点接收

到用户的查询后，负责查询解释，向其他节点请求信息，获取查询的结果。若是全部节点做出响应，那么回传给前端节点的大量响应产生碰撞，从而出现响应暴问题，如图 12-10 (f) 所示。信息采集机制的目的是响应质量最佳，即响应和响应数量最佳，同时网络资源消耗最少。

采用三种方法来完成信息采集任务：采样操作、自协调操作、扩散计算操作。

1. 采样操作

对于特定类型的应用（如确定整个网络区域的平均温度），每个传感器节点的响应可能产生响应暴问题。为了减轻响应暴问题，有些传感器节点若是其相邻节点做出响应则不必做出响应。传感器节点根据给定的响应概率自动决定是否应该参与应用，如图 12-10 (g) 所示。

假如传感器节点不是均匀分布在区域中，那么可以改进上述方法。为了防止密集区域产生过多响应，每个群首节点根据每个分群所要求的应答数量计算响应概率，将这种操作称为自适应概率响应（Adaptive Probability Response, APR）。

2. 自协调操作

在节点数较少的网络中，所有节点响应查询对于最终结果的精确度很重要。防止响应暴的另外一种方法是每个节点将其响应发送推迟一段时间。这种方法尽管会增加一些时延，但是能够降低碰撞机会，从而提高总体性能。

假定节点均匀分布在网络地理区域中，因此远离前端节点 h 个转发跳的节点数量正比于 h^2 。每个节点的响应时延定义为 $T_{\text{delay}}=KH(h^2-(h-1)r)$ ，其中 r 表示随机数 ($0 < r \leq 1$)， H 为每个转发跳时延估计常数， K 是一个补偿常数，补偿因子 K 是为了考虑排队时延和处理时延的影响， K 和 H 通常作为联合可调参数。

3. 扩散计算操作

对于扩散计算操作，假定每个传感器节点只知道其直接相邻节点。信息采集算法受到每个节点只能与其邻近区域内节点通信的限制。在传感器节点间分发 SQT 脚本中编写的信息累积逻辑，使这些传感器节点知道如何累积传输途中传递给前端节点的信息。图 12-10 (h) 描述了一个概念数据流。因为数据在传输途中被中间节点累积，所以有效带宽的占用明显减少，响应暴问题得到减轻。但是，对于大规模传感器网络，扩散计算操作可能需要较长时间才能将响应回传至前端节点。

SINA 采用分层结构，对于同一个应用，不同的层可以采用不同的信息收集方法，优化系统总体性能。在 SINA 中可以采用 SPIN 协议，SPIN 协商过程能够减少带宽的使用。通过将 SPIN 和 SINA 综合在一起，SINA 能够进一步节省网络资源。

12.4.6 应用举例

下面介绍传感器网络车辆跟踪应用及其 GloMoSim 仿真，说明 SINA 适用于传感器网络的查询与任务分配。对仿真传感器网络进行如下假设：

- 所有传感器节点固定不动，传感器网络不会发生分割；
- 传感器节点是同类型传感器节点，能力相同；
- 所有通信是对称通信；
- 在算法执行期间不会发生传感器节点失效；
- 传感器网络没有网络层提供的路由支持，但是，应用能够跟踪发送传感器的地址，并将该发送节点作为回传结果的接收节点。

传感器网络车辆跟踪应用就是对特定车辆或者移动目标进行定位、监视其运动。为了检测和识别一个目标，可能需要综合多种类型传感器的感知数据（如摄像机的图像、地震传感器的振动数据、音频传感器的噪声等）。对这些结果进行处理，并与感兴趣目标的特征进行比较。这里的主要兴趣是按照 SCTL 脚本形式编写协调算法程序，并将该程序分发给传感器节点执行。脚本控制各个传感器节点联合、有效、高效地检测感兴趣目标的出现。假定传感器节点能够根据联合感知信息的处理获取被跟踪车辆检测和识别的最终处理结果。

采用普通车辆跟踪法跟踪移动目标，即要求每个传感器节点同时感知和检测移动目标的特征。这种方法可能会浪费传感器节点的处理时间，因此没有高效利用传感器网络的有限能量，缩短了整个网络系统的寿命。

图 12-11（a）给出了协调车辆跟踪算法，该算法以抑制和重新初始化机制为基础，达到良好的跟踪效果，但是网络资源消耗低于普通车辆跟踪法。协调跟踪算法的主要原理是第一个检测到车辆的传感器节点抑制所有其他传感器节点的感知活动，因此其他传感器节点作为旁观者而可能不参与跟踪活动，从而节省能量。第一个传感器节点必须重新初始化其相邻节点的感知活动，保持对移动车辆的连续跟踪。只要车辆的行驶速度不大于重新初始化消息的传播速度，传感器网络就能够保持对移动车辆的连续跟踪。跟踪过程如图 12-11（b）所示。

（1）仿真建立

通过仿真比较协调车辆跟踪算法、普通车辆跟踪法的效率。仿真网络类似于诊断应用传感器网络，但是栅格单元 200 m，电台传输距离 380 m，网络覆盖区域 6 800 m×6 800 m，每个传感器节点具有跟踪和识别 200 m 范围内移动目标能力。当仿真网络开始运行时，一辆汽车从位置（5，5）开始以速度 15 m/s、沿着直线朝位置（6 800，6 800）行驶。两个跟踪应用从 15 s 开始运行、持续 10 min。跟踪频率为 7.5 次/s，或者传感器节点每隔 8 s 探测一次移动目标。对于协调车辆跟踪算法，一个传感器节点被抑制后，随后通过重新跟踪消息又接收到一个重新初始化，则重新启动其感知能力。在重新跟踪状态，一个传感器节点若是连续 40 s（重新跟踪周期）没有检测到移动目标，则停止感知移动目标，以便节省能量。

（2）仿真结果及其分析

表 12-7 给出了协调车辆跟踪算法、普通车辆跟踪法的仿真结果。采用有用感知与总感知之比率评估两个算法跟踪和监视移动目标的效率。有用感知定义为成功检测到车辆的感知。实验发现：两种算法的有用感知数量非常相同（209 次），但是两种算法的总感知数量差别很大（普通车辆跟踪法的总感知 76 800 次，协调车辆跟踪算法的总感知只有 8 828 次），这是因为普通车辆跟踪法缺乏传感器节点之间的协调。比较的第二个指标是在整个仿真期间所有节点发送的分组总数量，从表 12-7 中看到协调车辆跟踪算法使用的网络带宽多于普通车辆跟踪法，前者增加的分组全是跟协调有关的分组，即抑制消息、重新跟踪消息。但是，当比较操作总开销时，协调车辆跟踪算法优于普通车辆跟踪法：按照感知开销（ C_s ）与发送开销（ C_t ）比率 $C_s:C_t=4:1$ 计算总开销，然后将普通车辆跟踪法总开销标准化为 1，实验结果表明协调

车辆跟踪算法的总开销是普通车辆跟踪法总开销的 17.9%。

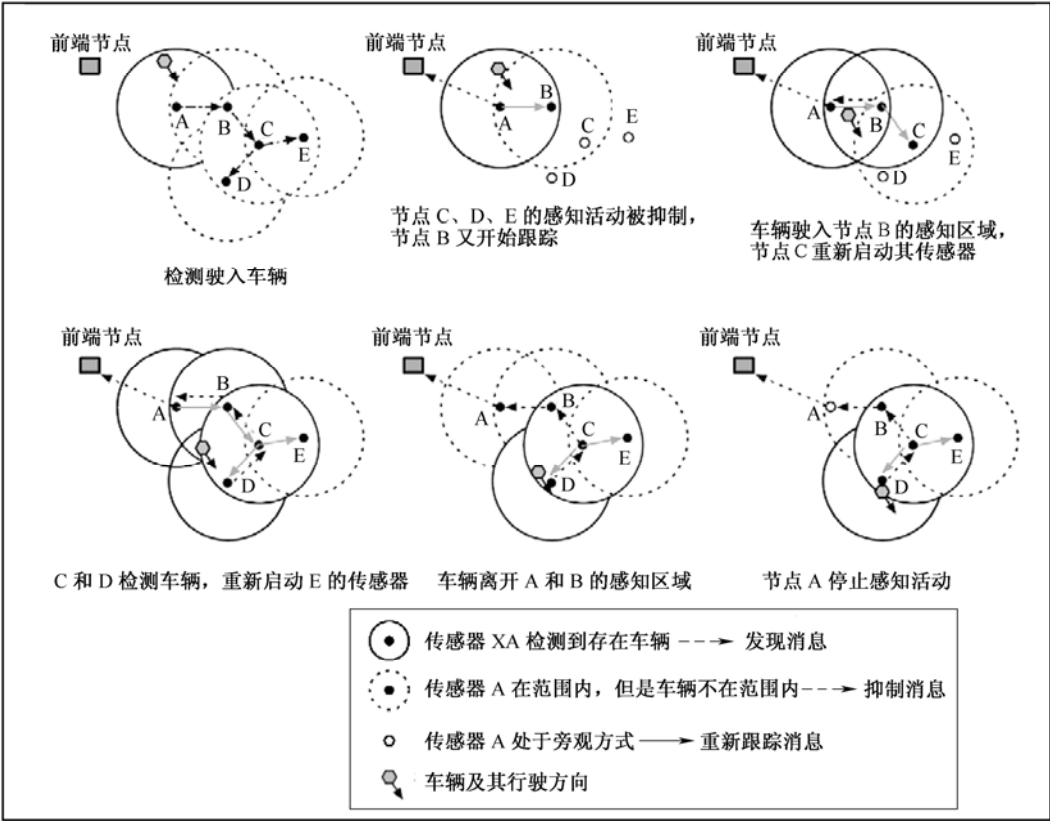
表 12-7 协调车辆跟踪算法、普通车辆跟踪法的仿真结果对比

车辆跟踪方法	有用感知与总感知之比	发送分组数量	标准化开销
普通车辆跟踪法	249:76 800 (1:308)	16 868	1.000
协调车辆跟踪算法	249:8 828 (1:35)	22 691	0.179

```
<execute>
  <sender> FRONTEND </sender>
  <receiver> <group>NODE0 </group>
    <criteria>TRUE </criteria>
  </receiver>
  <application-id>118 </application-id>
  <run-hop>0 </run-hop>
  <language>SQT </language>
  <with>
    <parameter type="clocktype" name="trackingTime" value="600" />
    <parameter type="clocktype" name="reTrackingTime" value="40" />
    <parameter type="clocktype" name="trackingFrequency" value="8" />
    <parameter type="object" name="target" value="Vehicle1" />
  </with>
  <content><[CDATA[
    lastSensingResult = false;
    timerApplication.start(); // instantiate a timer
    timerReTracking = createTimer(reTrackingTime);
    execute (ALL_NODES, "TRUE", MESSAGE["content"]); // re-broadcast
    if ((sensor1 = getMotionSensor()).turnOn()) { // instantiate a sensor object
      upon { // and turn it on
        receive (msg) where msg["action"] == "tell" && msg["content"] == "suppress"; {
          sensor1.standby(); break;
        }
        every (trackingFrequency); {
          if (sensor1.detect(target)) {
            tell (ALL_NODES, "TRUE", "suppress");
            tell (NEIGHBORS, "TRUE", "retrack");
            tell (MESSAGE["sender"], "TRUE", "found");
            lastSensingResult = true;
            timerReTracking.start();
            break;
          }
          else lastSensingResult = false;
        }
        expire (timerApplication); sensor1.turnOff(); exit(0);
      }
    }
  ]]> </content>
</execute>
```

```
upon { // After one sensor node sees the vehicle
  receive (msg) where msg["action"] == "tell" && msg["content"] == "retrack"; {
    if (timerReTracking.expired()) {
      sensor1.turnOn();
      timerReTracking.start();
    }
  }
  receive (msg) where msg["action"] == "tell" && msg["content"] == "found";
  tell (MESSAGE["sender"], "TRUE", "found");
  every (trackingFrequency); {
    if (sensor1.detect(target)) {
      tell (MESSAGE["sender"], "TRUE", "found");
      if (lastSensingResult) {
        tell (NEIGHBORS, "TRUE", "retrack");
        lastSensingResult = true;
        timerReTracking.start();
      }
      else {
        if (lastSensingResult) {
          timerReTracking.restart();
          lastSensingResult = false;
        }
      }
    }
    expire (timerReTracking); sensor1.standby();
    expire (timerApplication); sensor1.turnOff(); exit(0);
  }
} else exit(1);
]> </content>
</execute>
```

(a) 协调车辆跟踪算法的完整 SQT 代码



(b) 车辆跟踪过程

图 12-11 SINA 应用举例

参 考 文 献

- [1] Yu, Y. et al. Issues in designing middleware for wireless sensor networks. IEEE Network, Vol.18, No.1, pp.15-21, 2004.
- [2] R`omer, K. et al. Middleware Challenges for Wireless Sensor Networks. ACM Mobile Computing and Communication Review, October 2002.
- [3] Blumenthal, J. et al. SeNeTs - Test and Validation Environment for Applications in Large-Scale Wireless Sensor Networks. Proc. INDIN, 2004.
- [4] Demers, A. et al. The Cougar Project: a work-in-progress report. ACM SIGMOD, Vol.32, No.4, pp.53-59, 2003
- [5] Yao, Y. et al. The Cougar Approach to In-Network Query Processing in Sensor Networks. ACM SIGMOD, Vol.31, No.3, pp.9-18, 2002.
- [6] Shen, C.C. et al. Sensor Information Networking Architecture and Applications. E Personal Communications Magazine, Vol.8, No.4. pp.52-59, 2001.
- [7] Srisathapornphat, C. et al. Sensor Information Networking Architecture. Proc. Int. Workshops on Parallel Processing , 2000.
- [8] A. Cerpa, J. Elson, et al. Habitat Monitoring: Application Driver for Wireless Communication Technology. In ACM SIGCOMM Workshop on Data Communications, Apr. 2001.
- [9] J. Heidemann, F. Silva, C. Intanagonwiwat, et al. Building Efficient Wireless Sensor Networks with Low-Level Naming. In Proceedings of the 18th ACM Symposium on Operating Systems Principles, Oct.2001.
- [10] ATMEL. AT45DB041B, 4M bit, 2.7-Volt Only Serial-Interface Flash with Two 264-Byte SRAM Buffers data sheet. <http://www.atmel.com/>, June 2003.
- [11] P. Eggenburger. GPS-MS1E Miniature GPS Receiver Module Data sheet. <http://www.u-blox.ch/>, Oct. 2001.
- [12] J. Hill and D. Culler. Mica: A Wireless Platform for Deeply Embedded Networks. In Micro, IEEE, volume 22, pages 12–24, 2002.
- [13] P. Juang, H. Oki, Y. Wang, et al. Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet. In Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X), Oct. 2002.
- [14] T. Liu and M. Martonosi. Impala: A Middleware System for Managing Autonomic, Parallel Sensor Systems. In ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP '03), June 2003.
- [15] Liu, T. et al. Implementing software on resource-constrained mobile sensors: experiences with Impala and ZebraNet. Proc. MobiSys, 2004.
- [16] P. Levis and D. Culler. Mate: A Tiny Virtual Machine for Sensor Networks. In Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X), Oct. 2002.

- [17] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless sensor networks for habitat monitoring. In ACM International Workshop on Wireless Sensor Networks and Applications(WSNA'02), Atlanta, GA, Sept. 2002.
- [18] Maxstream. 9XStream, wireless modem data sheet and OEM manual. <http://www.maxstream.net/>, June 2002.
- [19] Mpala Wildlife Foundation. Mpala research centre. <http://www.mpalafoundation.org/researchctr/>.
- [20] Panasonic. CGR18650A, 2A-hour Lithium-Ion battery, cylindrical Model. <http://www.panasonic.com/>, Aug. 2003.
- [21] Sun Microsystems. Java 2 Platform, Micro Edition. <http://java.sun.com/j2me/>, Nov. 2002.
- [22] Texas Instruments. MSP430x1xx Family Ultra-Low-Power Micro-controller User's Guide. <http://www.ti.com/>, 2002.
- [23] The Center for Conservation Biology. VAFALCONS. <http://fsweb.wm.edu/ccb/vafalcons/falconhome.cfm>, 2002.
- [24] B. Bayerdorffer. Distributed Programming with Associative Broadcast. Proc. 28th Hawaii Int'l. Conf. Sys. Sci., Jan. 1995.
- [25] C. Jaikaeo, C. Srisathapornphat, and C.-C. Shen. Querying and Tasking in Sensor Networks. SPIE's 14th Annual Int'l. Symp. Aerospace/Defense Sensing, Simulation, and Control, Orlando, FL, Apr. 2000.
- [26] P. Bonnet, J. Gehrke and P. Seshadri. Querying the Physical World. IEEE Pers. Commun., vol.7, Oct.2000, pp.10–15.
- [27] C. Jaikaeo, C. Srisathapornphat, and C.-C. Shen. Diagnosis of Sensor Networks. IEEE ICC, Helsinki, Finland, June 11–14, 2001.
- [28] J. Byers and G. Nasser. Utility-Based Decision-Making in Wireless Sensor Networks. Tech. rep. BU-CS 2000-014, Comp. Sci. Dept., Boston Univ., June 2000.
- [29] Heinzelman, W. et al. Middleware to support sensor network applications. IEEE Network, Vol.18, No.1, pp.6-14, 2004.
- [30] Mark, A. et al. Providing Application QoS through Intelligent Sensor Management. Proc. SNPA, 2003.

第 13 章 无线传感器网络应用及编程

本书已经结合 WSN 网络技术介绍了一些 WSN 应用，下面对 WSN 应用进行概括，然后描述 WSN 应用设计原理，接着介绍 WSN 网络编程基本技术，最后将重点描述两个 WSN 网络编程技术——ATaG 编程体系架构和融合应用编程体系结构（DFuse）。

13.1 传感器网络的应用

起初 WSN 的研究是受到军事应用推动的。军用 WSN 的例子从用于海洋监视的大规模声学监视系统到用于地面目标检测的小型无人地面 WSN。但是，低成本传感器通信网络的实用性已经导致开发许多其他可能的应用，包括从基础设施安全到工业传感的新应用。

WSN 可能由许多不同类型的传感器组成，比如地震传感器、低采样率磁传感器、热传感器、视觉传感器、红外线传感器、声学传感器、雷达等，这些传感器能够监视以下各种周围情况：温度；湿度；车辆移动；雷电；压力；土壤结构；噪声大小；判定特定目标是否存在；所连接目标的机械压力；目标的当前特性（如速度、方向、大小）。

传感器节点可以用来连续感知、事件检测、事件 ID、位置感知以及本地控制。传感器节点的微型感知和无线连接产生许多新的应用领域，可将这些应用分成军事、环境、医疗卫生、家庭以及其他商业应用六个方面，也可以将这种分类进一步增加太空探测、化学处理、救灾等。

13.1.1 军事应用

WSN 是军事指挥、控制、通信、计算、情报、监视、侦查、目标（Command, Control, Communications, Computing, Intelligence, Reconnaissance and Targeting, C4ISRT）系统的一个完整组成部分。WSN 展开迅速、自组织、容错特性使得其成为军事 C4ISRT 中极富前途的感知技术。WSN 由密集布置的、可任意使用的低成本传感器节点组成，因此，有些节点被敌方毁坏不会影响军事作战，就好像一个传统传感器被毁坏，WSN 概念是用于战场的一种较好解决方法。

自“9·11”以来，一直在关心有关防恐的安全问题，有关 WSN 的研究包括检测和预防恐怖分子袭击。WSN 的有些军事应用包括监视友方的军队、装备、军火；战场监视；敌方军队和地形的侦查；目标瞄准；战场损失评估；核攻击、生物攻击、化学攻击（Nuclear, Biological, Chemical Attack, NBC）的探测与侦查。例如，在城区（如主要十字路口）布置放射性传感器，利用放射性物质检测恐怖分组袭击。军用 WSN 能够检测有关敌方运动、爆炸以及感兴趣的其它现象，使用声学传感器能够计算狙击手的位置。美国的许多 WSN 研究计划（如 SensIT^[4]）得到了 DARPA 的支持。

① 友方军队、装备、军火的监视：领导和指挥官采用 WSN 能够连续监视战场上友方军

团的状态、装备和军火的情况和有效性。可以将小型传感器连接到每个军团、每部车辆、每个装备、重要军火，监视和报告其状态。中心节点收集这些报告信息，然后将其发送给军团的领导。也可以将这些报告信息与来自其他单元的数据累积在一起，再发送给指挥体系的上级。

② 战场监视：采用 WSN 能够迅速覆盖主要地带、行军路线、通道和航线，密切监视敌军的活动。随着军事行动的推进以及新作战计划的准备完毕，可以随时运用新 WSN 进行战场监视。

③ 敌方军队和地形的侦查：可以将 WSN 布置到主要地带，在敌军能够拦截 WSN 之前的数分钟内及时收集一些有关敌方军队和地形的有价值的详细敌军情报。

④ 目标瞄准：可以将 WSN 综合到智能军事装备的引导系统中，实现目标瞄准。

⑤ 战场损失评估：刚好在攻击前或攻击后将 WSN 布置到目标区域，用于收集战斗损失评估数据。

⑥ 核攻击、生物攻击、化学攻击的检测与侦查：在化学生物战争中，接近地面零点对于及时、准确地检测有关物质非常重要。布置在己方战场用做化学生物告警系统的 WSN 能够给己方部队提供临界反应时间，大幅度降低部队的伤亡。在检测到核攻击、生物攻击、化学攻击后，也可以使用 WSN 来进行侦查。例如，采用 WSN 可以在侦查人员不暴露在原子弹辐射下就能够进行原子弹侦查。

⑦ 基础设施安全：WSN 可用于基础设施安全和反暴。诸如电厂、通信中心之类的重要建筑和设施必须受到保护，使其免受恐怖分子的袭击。可以在这些重要建筑和设备周围布置电视视频网络、声控网络以及其他 WSN。这些传感器能够及早检测出可能存在的威胁。通过融合多个传感器的数据，能够改善覆盖范围、检测能力以及降低虚假告警率。尽管与固定网络连接固定传感器能够保护大多数设备，但是无线移动 Ad Hoc 网络灵活、机动性较强，在需要之时能够提供额外的覆盖范围。此外，WSN 还可用于环境监视，检测建筑物变形问题和结构问题，防止建筑灾祸。

13.1.2 环境应用

WSN 可以用于检测和监视平原、森林、海洋、洪水、精密农艺等中的地域性环境变化。这些应用的一个共同特点是：在服务器累积传感器数据，将累积结果以及位置和其他环境信息分发给用户。例如，GlanceWeb^[4]监视冰河，观测冰河的变化（可能由全球变暖引起）；Burrell^[6]使用传感器控制葡萄园，记录和分析农民耕作期间的变化。Zhang^[10]开发了一个 ZebraNet，利用传感器和 ZebraNet 观测和跟踪野生动物（详见第 12 章）。

WSN 的有些环境应用还包括鸟类飞行、小动物爬行、昆虫飞行的跟踪，监视影响农作物和家畜的环境条件，用做大规模地球监视和行星探测的微型装置，化学/生物探测；精密农艺；根据海上、土壤、大气进行生物、地球、环境监视；森林大火探测；环境的生物复杂性测绘；污染研究。

① 森林火灾探测：因为传感器节点可以有策略地、任意地、密集地布置在深林中，所以传感器节点能够在火势蔓延到无法控制之前将准确的火源信息中继传输给端用户。可以布置数百万个传感器节点，全部采用无线/光系统。此外，可以给传感器节点设置有效的功率提取法，比如太阳能电池，这是因为传感器节点可以无绳工作数月甚至几年时间。各个传感

器节点相互协作，共同执行分布式感知任务，克服障碍物（如树木、岩石），障碍物阻碍有线传感器的视距通信。

② 环境的生物复杂性测绘：环境的生物复杂性测绘需要复杂方法来综合随着时间和空间而变化的信息。远距离感知与自动化数据收集技术的进步已经使人们能够以每单位面积几何下降的成本寻求更好的空间、频谱、时间的解决方案。随着这些技术的进步，传感器节点也将能够连接互联网，允许远端用户控制、监视、观察环境的生物复杂性。

卫星和航空传感器适用于观测宽广的生物多样性（比如支配植物种类的空间复杂性），但是不够精细，不能观测生态系统中的微小生物多样性。因此，需要在地面布置无线传感器节点，用于观测生物多样性。例如，在南加利福尼亚州詹姆士预备队研制了环境生物复杂性测绘，实现了三个监视栅格网，每个栅格网由 25~100 个传感器节点组成，用做固定观测多媒体和环境传感器数据记录器。

③ 洪水监测：洪水监测的一个例子是美国开发的 ALERT 系统^[16]。ALERT 系统中布置了三类传感器：降雨量传感器、水位传感器、气候传感器。这些传感器按照预定方式给中心数据库系统发送信息。许多研究项目，比如美国康奈尔大学的美洲豹装置数据库项目、Rutgers 的数据库项目^[5]，研究了传感器场中传感器相互间的分布式交付方法，以便提供快速查询和长期查询。

④ 环境与栖息地的监视：环境与栖息地监视是 WSN 的一种当然应用，这是因为被监测的变量（比如温度）通常分散在一个大区域中。例如，最近起建于美国洛杉矶的嵌入式网络传感中心^[13]重点研究环境与栖息地监视问题。环境传感器用于研究植物对气候趋势和疾病的反应，声学传感器、成像传感器用于识别、跟踪、测量鸟群和其他物种的数量。至于大系统，比如巴西政府发起的亚马逊警戒系统（System for Vigilance of the Amazon, SIVAM）^[14]提供亚马逊盆地的环境监视、毒品交易监视、空中交通控制，这个庞大的 WSN 由相互连接在一起的不同类型的传感器组成，包括雷达、成像传感器、环境传感器。成像传感器是空间传感器，雷达安装在航空器上，大多数环境传感器布置在地面。连接传感器的通信网络按照不同速度工作。例如，高速网络将传感器连接到卫星和航空器上，而低速网络连接地面传感器。

⑤ 精密农艺：有些好处是能够实时监视饮用水中农药浓度、土地腐蚀程度、空气污染程度。

13.1.3 医疗卫生应用

WSN 在医疗健康方面的应用包括远程监视人体生理数据、跟踪和监视医院内的病人和医生、辅助老人、联系伤残人员、病人综合监视、诊断、医院药物管理、监视昆虫或者其他小动物的运动和内部过程。可佩戴传感器、可植入传感器能够连续不停地监视病人的各种状况，从而能够缩短获取病人检查结果所需时间，对病人恢复快慢有直接关系。对于挫伤病人，医师作出迅速而精确诊断以及正确医治方法特别重要。远程监视也能够进行远程治疗检验、在远端位置上开始治疗，辅助事故地点、灾害地点的精确定位。

① 人体生理数据的远程监视：通过 WSN 收集到的生理数据可以存储较长时间，可以用于医学研究。WSN 可以监视和察觉老人的行为，比如摔跤。小型传感器节点使得人们具有较大自由移动度，允许医生提前识别预定的症状，有助于提高人们生活质量（相对于治疗中心）。法国东南部的格勒诺布尔市医学院设计的“健康智能之家”验证了这种系统的可行性^[17]。

② 医院内部医生和病人的跟踪和监视：每个病人配备微小轻型传感器节点。每个传感器节点有其专门的任务。例如，一个传感器节点用于探测心脏跳数，另一个传感器节点用于探测血压。每个医生也可以配备一个传感器节点，以便其他医生确定自己在医院的位置，比如可以参阅第 10 章介绍的 DFA、第 12 章介绍的 MiLAN。

③ 医院药物管理：假如传感器节点能够与药物治疗连接，那么错误治疗病人、给病人开错药的机会可以降到最低程度。这是因为，病人配备的传感器节点能够确定其反应和所需要的治疗。参考文献^[18]描述的计算机系统已经表明：WSN 能够帮助实现将错用药物事件降到最低程度。

13.1.4 家庭应用

家庭自动化和环境智能的目标是使用大量网络化传感器创建智能空间。Smart-Its^[20]研究项目是在日用商品中使用传感器，利用自治 WSN 替代必须人工交互的条形码，自动检测货物的位置和质量（温度、湿度），实现自动化管理系统。存货控制管理、车辆跟踪、交互式博物馆均属于 WSN 智能化应用领域。

① 家庭自动化：随着技术的发展和进步，能够将灵敏传感器节点潜入到家用装置中，比如吸尘器、微波炉、电冰箱、录放机、VCD、DVD 以及 VCR。嵌入到家用装置中的传感器节点能够相互交互，以及通过互联网或者卫星与外部网络连接，使端用户更加易于本地和远程管理家用装置。

② 环境智能：环境智能设计包括两个不同方面：以人为中心的环境智能和以技术为中心的环境智能。对于以人为中心的环境智能，环境智能必须自适应端用户输入/输出能力的需求。对于以技术为中心的环境智能，必须开发新的硬件技术、网络解决方案、中间件服务。参考文献[2]描述了使用传感器节点来建立智能环境的一种设计方案。传感器节点可以嵌入到家具、器具中，能够相互通信，也能够与室内服务器通信。室内服务器能够与其他室内服务器通信，以便获知其他室内服务器提供的服务，比如打印、扫描、传真等。室内服务器和传感器节点可以与现有嵌入式装置综合在一起，根据参考文献^[2]所描绘的控制论模型形成自组织、自控制、自适应系统。另外一个例子是美国佐治亚州理工学院开发的“住宅实验室”^[25]。智能环境必须具有可靠、持久、透明的计算和感知。

13.1.5 其他商业应用

有些商业应用是：材料疲劳度监视；虚拟键盘建立；存货管理；产品质量监视；智能化办公室建设；办公大楼的环境控制；自动化生产环境的机器控制和指导；与玩具的交互；与博物馆的交互；工厂生产加工过程的控制与自动化；灾区监视；传感器节点嵌入式的结构智能化；机器诊断；运输；工厂仪器设备；传动装置的本地控制；盗车的检测与监视；车辆跟踪与检测；半导体处理室、旋转机器、风洞、消声室的仪器设备。

① 办公大楼的环境控制：大多数办公大楼的空调和热度可以集中控制。因此，房间内的温度可以变成只有几度；房间内只有一个控制，来自中央系统的空气流不是均匀分散的，所以房间内一边的温度高于另一边的温度。可以安装分布式 WSN 系统来控制房间内不同地方的空气流和温度。据估计，这种分布式技术能够降低能耗约 $2 \times 1\,024$ BTU，在美国，相当

于每年节省 550 亿美元, 减少 35 000 000 吨碳的发热量^[27]。

② 交互式博物馆: 将来孩子们能够与博物馆中展物进行交互, 从而更多、更深入地了解展物。展物能够响应孩子们的触摸和声音。此外, 孩子们可以参与实时因果试验, 试验教给孩子们有关科学和环境的知识。而且, 传感器节点可以在博物馆内提供寻呼和定位服务。例如, 旧金山探险博物馆将数据测量和因果试验综合在一起^[27]。

③ 汽车盗窃的检测与监视: 可以在某个区域内布置传感器节点, 用于检测和确定该区域内的盗车情况, 并通过互联网将盗车情况通知远处的端用户, 以便进行案情分析。

④ 货仓存货控制的管理: 一个仓库中的每件货物可以安装一个传感器节点。端用户可以找出每件货物的精确位置、统计同类货物的数量。假如端用户需要存入新货物, 那么所有用户需要做的就是给新加货物系上合适的传感器节点。端用户可以一直跟踪任何仓库中的货物以及确定其中每件货物的位置。

⑤ 车辆跟踪与检测: 参考文献[28]讨论了两种车辆跟踪与检测方法: 第一种方法是就地确定各个汽车组的车辆方位线, 然后将该信息转发给基站; 第二种方法是将传感器收集的原始数据转发给基站, 以便基站确定车辆的位置。

⑥ 工业检测: 商业、工业界长期感兴趣将传感器作为降低成本、提高机器(或许包括用户)性能和可维护性的一种手段。通过确定机器的震动或者磨损程度以及润滑油量来监视机器的“健康”, 将传感器安插到人难以接近的区域, 就是两个传感器工业应用的例子。IEEE 和美国国家标准化技术协会(NIST)发布了 P1451 灵敏传感器接口标准, 以便能够在工业环境中采用即插即用传感器和网络。工厂一直在持续不断地运用远端传感网络进行自动化生产和装备线, 采用传感器实现复杂的在线质量控制测试。尤其可以给工厂装备远端无线传感器, 以确保遵从联邦安全指导方针, 同时又保持低安装成本。

光谱传感器也用于工业环境。从简单的光学装置(比如 pH 探测器)到能够起微型分光计作用的真正光谱装置, 光学传感器能够替代现有仪器完成物质属性、组成成分的测定。由于低成本电荷耦合装置(Charge Coupled Device, CCD)阵列装置和小型化工程能够使传感器体积变得越来越小、能力却越来越灵敏, 所以小型化推动光学传感器的发展。各种各样工业传感器的目的就是为了进行多点传感或者矩阵传感: 将数百个、数千个传感器的输入信息存入到数据库中, 数据库有许多方法访问, 以便表示大范围或者小范围内的实时信息。

⑦ 交通控制: WSN 一直用于交通车辆监视和控制。大多数交通十字路口都有空中传感器或者地下传感器, 用于检测车辆和控制交通信号灯。此外, 经常采用摄像机来监视交通繁重路段, 将交通繁重路段的视频信息发送给中心的操作员。但是, 这些传感器和与其连接的通信网络成本太高, 因此, 通常在几个关键点设置交通监视。廉价的无线 Ad Hoc 网络将彻底改变交通监视和控制的面貌。可以在每个路面十字路口布置具有嵌入式网络能力的廉价传感器, 进行车辆监视和计数以及估计车辆的速度。这些传感器与相邻节点通信, 最终得到一张“全局交通图”, 操作员、自动控制器可以查询这张图来产生控制信号。

另外一个基本概念是将传感器安装到每辆车上。车辆彼此通过的时候, 就可以交换以下全部信息: 交通堵塞地点、行驶速度、交通密度, 以及地面传感器可能产生的信息。这些信息从一辆车传播到另一辆车, 驾驶员运用这些信息来避开交通堵塞路线、选择其他不堵塞的行驶路线。

13.2 WSN应用设计原理

WSN 应用设计需要考虑传感器布置、移动性、基础设施、网络拓扑、网络密度、网络规模、网络连通性、网络寿命、节点寻址、数据累积、查询能力、查询分发、感知数据分发、实时性、可靠性、自构、安全等方面的问题，并据此确定 WSN 的操作方式。

13.2.1 设计方面

WSN 的基本目标主要依赖应用。当在使用 WSN 时，应用不是孤立的，而是被综合到较大的计算基础设施中。

WSN 研究的困难在于建立一个能够容纳各种物理现象的开放系统。必须考虑硬件约束条件，比如能力与功耗之间的关系。为了处理 WSN 研究的复杂性，Estrin 等人提出了两个设计原理^[29]。一是数据中心设计，重点在于传感器数据的管理，而不在于传感器节点的管理；二是特定应用设计方法，设计应用环境的物理特征和社会特征。

下面分析应用特征的各个方面，目的是帮助 WSN 应用设计。对 WSN 应用采用正确的设计方法和技术非常重要。

1. 布置

传感器节点可以布置在特定位置上，也可以随机布置。完成初始布置后，可能需要增加或者替换传感器，这会影响节点位置、节点密度以及总的拓扑结构。

每个传感器节点的程序可以人工加载，也可以运行时加载。

2. 移动性

传感器节点可以安装在移动实体上。移动性或者是附带结果，或者是系统所要求的属性（比如将传感器节点移动到感兴趣的物理位置上）；移动性要么是主动的（比如汽车），要么是被动的（比如连接到移动目标上，但是又不在传感器节点的控制下）。移动性可应用于所有网络节点或者一部分网络节点。移动程度也可能变化，从长时间固定不动伴随偶尔移动到持续不停的移动。移动性对预期的网络密度动态性影响甚大，因此影响网络协议和分布式算法的设计。实际移动速度也可能有影响。

3. 基础设施

可以采用不同方法运用各种通信形式建立通信网络。两种常用形式是：基于基础设施的网络，Ad Hoc 网络。在基于基础设施的网络中，传感器节点可以直接与基站通信，基站的数量取决于传感器节点的通信距离和覆盖区域。在 Ad Hoc 网络中，传感器节点之间可以直接通信，不需要基础设施，传感器节点可以作为路由器为其他传感器节点进行消息多跳转发。

应该考虑成本。当前技术发展水平已允许使用蓝牙无线系统，其成本低于 10 美元。要实现 WSN 切实可用，一个传感器节点的成本应该远低于 1 美元。

4. 网络拓扑

WSN 的一个重要特性是其直径（即网络中任意两个节点之间的最大转发跳数）。最简单的形式就是 WSN 是一个单跳网络，每个传感器节点能够直接与任何其他传感器节点通信。具有一个基站的基础设施网络构成一个星状网络，直径等于两个转发跳。多跳网络可以构成任意图，但是经常是构成结构较简单的覆盖图网络（比如树或者一个连通星状结构组成的集合）。拓扑影响许多网络特征，比如时延、强壮性、容量，数据传输路由以及数据处理的复杂性也跟拓扑有关。

假如大量节点布置在不同位置上，那么网络必须能够自组织，人工配置行不通。节点可能失效（或者由于没有能量，或者由于物理破坏），所以需要在网络中增加新的节点。因此，网络必须能够周期性自行重构。单个传感器节点可能失去与剩余网络的连接，所以必须维持高密度的连通性。

5. 网络密度与网络规模

由一个传感器节点的有效传输距离定义该节点的覆盖区域。节点密度（其计算公式参阅第 1 章）表示传感器节点覆盖一个感兴趣区域的覆盖程度。网络规模影响可靠性、精确度、数据处理算法。一个直径不足 10 m 的区域的节点密度可以从少数几个传感器节点到上百个传感器节点。

6. 连通性

由单个传感器节点的通信距离和物理位置定义网络的连通性。假如任意两个节点之间总是存在网络连接（可能是多跳连接），那么就说网络是连通的。假如网络偶尔发生分割，那么连通性是断断续续的。假如节点在大部分时间是孤立的，只是偶尔进入其他节点的通信覆盖范围内，就说是偶发通信。连通性影响通信协议和数据分发机制的设计。

7. 寿命

WSN 所要求的寿命跟应用有关，从几个小时到数年之长。必需的 WSN 寿命对所要求的能量效率和节点强壮性影响很大，因此要求能耗最低。

8. 节点可寻址

节点可寻址表示节点是否可以单个可寻址，这与应用有关。例如，停车场网络中的传感器节点应该单个可寻址，这样才能够确定所有空闲车位的位置。因此，可能必须将消息广播给网络中的所有节点。假如需要确定一个房间某个角落的温度，那么可寻址就没有车位应用那么重要。给定区域内任何节点都可以做出响应。

9. 数据累积

数据累积就是对在 WSN 中传输的数据进行概括和提炼。太多传感器节点泛洪信息容易引起网络拥塞，这个问题的主要解决方法就是累积或者融合 WSN 内的数据，然后将累积数

据发送给控制器。

有三种主要的数据累积方法：第一种方法是扩散算法假定数据从一个节点发送给下一个节点，因此数据在网络中传播、直至到达目的地，数据在传播期间可能被累积（大都采用简单函数，并且假定是同类型数据）；第二种方法是连续查询，以扩充 SQL 为基础的流查询，数据是短暂的，而查询是持续的；第三种方法是事件图影响事件流，根据事件代数学将一个的一个的单个事件组合成累积事件。对反应式中间件的事件代数学扩充了 WSN 事件相关的时间约束条件和发送概率。按照事件消耗方式使用事件。

上述三种累积方法影响 WSN 中数据累积的时间和地点，涉及到分布式不可靠环境中的时间处理问题、状态、异步不稳定通信、冗余度等。不同的累积机制要求不同的资源，因此影响路由策略，比如路由适应应用或者应用适用路由。必须决定是否在传感器节点或者在资源比较丰富的专门节点进行查询处理，是否在外围感知节点放置简单滤波器。典型的累积操作包括 MAX、MIN、AVG、SUM 以及其他许多众所周知的数据库管理技术。

10. 查询能力与传播

WSN 中存在两种类型的寻址：数据中心寻址和地址中心寻址。在数据中心寻址中，将查询发送给 WSN 的特定区域；在地址中心寻址中，将查询发送给单个传感器节点。用户在收集 WSN 某个区域的信息时可能需要查询其中某个传感器节点或者其中一组传感器节点。根据执行的数据融合数量，可能不能在网络中发送大量数据。各种各样的本地中心节点收集一个给定区域内的数据，产生概括性消息。可以将查询发送给所需位置附近的中心节点。

11. 数据分发

WSN 的最终目标是检测传感器场中感兴趣的特殊事件，并将其交付给用户。由于传感器邻近范围相互重叠，所以相同物理现象可能会被多个传感器节点所记录。系统累积可能会丢失相同物理现象的所有数据。需要符合 WSN 特点的端到端事件传输协议，有线分布式系统同样需要异步通信交付语义，比如发布/订阅。

功耗与传感器数据的处理方法和通信方法密切相关。由于传感器节点的电池能量非常有限，所以必须考虑应用需求所能够满足的程度，可以采用自适应通信协议（功率意识协议）。

12. 实时性

目标跟踪应用可能需要对来自不同源节点的事件进行实时相关性处理。实时性支持（比如必须在一定的时间周期内报告物理事件）对于 WSN 可能非常重要，影响时间同步算法，时间同步算法受网络拓扑和所采用的通信机制的影响。

13. 可靠性

可靠性可用泊松分布来模拟（第 1 章），容错等级依赖 WSN 应用。

14. 自组织

假定大量传感器节点扩散在敌方位置上，那么网络必须能够自组织。传感器节点可能由

于能量限制、物理破坏或者其他手段而失效，因此可能需要在网络中增加新的传感器节点。各个传感器节点能够相互协调，利用高密度提供的冗余度延长系统总寿命。这种系统采用大量传感器节点排除了人工配置，环境动态性排除了设计时的预配置。传感器节点必须自构，建立苛刻能量限制条件下的通信网络拓扑。**WSN** 必须能够周期性地、连续不停地进行自行重构，这样才能够连续不停地发挥正常作用、提供服务。高连通性密度是必要的。

15. 安全

WSN 的安全威胁包括：

① 被动信息采集：假如传感器节点之间的通信、传感器节点与中间节点或者数据收集点之间的通信采用明数据（未加密）通信，那么入侵者采用合适的接收机以及精心设计的天线就能够被动收集数据流。

② 颠覆节点：一个被捕获的传感器节点可能被篡改，因此极有可能失密。这个传感器节点一旦失密，就可能暴露其加密密钥信息和访问较高级通信，攻击者可以利用传感器功能进行攻击。因此，必须设计安全的传感器节点，能够预防哄骗，在发送时不会泄漏敏感信息。

③ 伪造节点：入侵者可能对 **WSN** 增加其自己的节点，输入虚假数据，堵塞真消息传输。一般地，虚假节点计算能力较强，能够假扮成某个传感器节点。尽管一直在研究分布式系统、**Ad Hoc** 网络中的这个问题（含恶意主机），但是其解决方法（组密钥协议、法定数协议、每个转发跳认证协议）一般计算量较大，不适用于 **WSN**。

④ 节点功能障碍：一个 **WSN** 节点可能发生功能障碍，产生不精确数据，甚至虚假数据。假如这个节点作为中间节点，为其他节点转发数据，那么这个节点可能丢掉或者混淆从其传输通过的分组。因此必须从 **WSN** 中检测和精选这种节点。

⑤ 节点停止工作：假如一个节点作为中间节点、数据收集点或者数据累积点，那么这个节点停止工作会造成什么后果？**WSN** 协议必须足够强壮，在面对节点停止工作时能够提供其他路由，减轻由此造成的不利影响。

⑥ 消息受损：针对消息完整性的攻击就是入侵者在源节点和目的地之间插入自己，并且篡改消息内容。

⑦ 拒绝服务：对 **WSN** 的拒绝服务攻击可以采用若干种形式。比如干扰无线链路、资源穷尽干扰、错误数据传输路由。

⑧ 流量分析：尽管通信可以加密，但是因果分析、通信模式与传感器活动可能会暴露足够信息，使得攻击者能够挫败或者暗中破坏 **WSN** 承担的任务。未加密保护而发送的寻址信息和路由信息也有利于攻击者进行流量分析。

13.2.2 确定WSN操作坊式

WSN 的操作（包括数据处理）需要考虑上述各个设计方面的复杂综合。传感器节点的操作多样化，具体依赖应用和传感器节点的使用。假定基站或者中心节点作为 **WSN** 的一个组成部分，那么 **WSN** 的操作分成以下几类：单跳操作方式；多跳操作方式；按需操作方式；自组织方式；数据累积方式；反应式处理方式。下面加以详细描述。

① 单跳到达中心节点：传感器节点感知数据，并将感知数据发送给中心节点（如基站、群首），中心节点位于传感器节点的传输覆盖范围内，因此不需要路由，也不需要传感器节点

之间的协调。所以这种操作方式使用集中式、点到点、单跳通信模式，就好像有基础设施支持的无线网络一样。这种操作方式存在的一个问题是传感器节点处在中心节点的传输覆盖范围之外时就不能进行通信。

② 多跳到达中心节点：多跳操作方式就是远离中心节点的传感器节点将其感知数据发送给相邻传感器节点，再由后者将感知数据转发给中心节点。转发过程涉及源节点与数据收集点之间传输路径上的多个传感器节点。因此，这种操作方式采用集中式多跳通信模式。不管传输路径有多长，数据最终会传递到达收集点。位于到达基站的传输路由上的传感器节点间的相互协调是这种操作方式的组成部分。

③ 按需操作：在按需操作中，传感器直接接收或者通过多跳转发接收控制器（比如基站或者中心节点）的命令，根据请求完成自行配置。单跳操作方式和多跳操作方式都是多点到一点通信模型，只是专门为非按需数据传输设计的。但是在按需操作中，可以从控制器（比如基站、中心节点）向整个 WSN 广播命令，也可以采用单目标传输方式将命令传输给若干个传感器节点（一点到多点通信模型）。WSN 中的数据传输代价极高，非按需数据传输有可能导致 WSN 寿命大幅度缩短。此外，有些传输是不必要的，比如 100 个传感器节点直接向中心节点报告某个区域的温度为 35℃。假如可以适当调整传输，那么可以延长 WSN 寿命，同时又不会影响到达中心节点的信息的质量。

例如，考虑使用一组传感器节点来监视温度。这组传感器节点布置完毕后开始按照空闲方式工作，这是低功耗方式。控制器向节点组广播苏醒消息，传感器节点接收到苏醒消息后做出响应，由空闲方式转换到活动状态。随后控制器广播提取数据命令，请求传感器节点的感知数据。最后，控制器命令传感器节点进入空闲方式。依次重复上述过程。因此，在进行非按需数据传输时，联合使用多点到一点通信模型和一点到多点通信模型的能量效率高于只使用多点到一点通信模型的能量效率。假如采用单目标传输方式传输消息，那么需要某种方法寻址每个单独的传感器节点，但是 WSN 节点既没有到达单个接收节点的路由，也没有 WSN 拓扑信息，因此不能保证单目标消息能够真正到达预定接收节点。

④ 自组织：WSN 布置完后，自组织，中心控制器知道网络拓扑。控制器（比如基站、中心节点）维护拓扑信息，并与部分或者所有 WSN 节点共享拓扑信息。自组织操作方式可能会使用能力较强的传感器节点作为 WSN 内一小部分节点的群首，要求其路由意识强。这种操作方式通过引入自组织概念延伸双向通信模型，包含三个主要任务：节点寻找、路由建立和拓扑维护。完成这三个任务即构成一个真正的 WSN。单跳、多跳和按需三种操作方式使用集中式通信模型，而自组织操作方式和下面介绍的数据累积、反应式处理两种操作方式联合采用集中式通信模型和分布式通信模型，使 WSN 尽可能高效地完成工作。WSN 拓扑维护不同于任何其他无线网络的拓扑维护。在 WSN 中，传感器节点可以固定不动，也可以移动。当 WSN 节点固定不动时，一旦建立起拓扑，那么拓扑通常不会发生变化。但是 WSN 节点执行其所承担的任务，消耗能量，最终会耗尽其全部能量，从而失效不再工作。因此，需要定期刷新 WSN，增加新的传感器节点。

⑤ 数据累积：数据累积操作方式就是执行数据累积，减小流量，节省能量。所有传感器节点采用按需方式或者非按需方式朝基站（或者中心节点）发送数据。

⑥ 反应式处理：上述 5 种操作方式必须考虑数据收集，并将其交付给中心节点。WSN 节点不参与通过感知任务获得的数据的语义。依据输入数据和自己的测试结果作出判断要求传感器节点根据这些数值进行操作。操作可以是计算或者输入数据触发的操作。假如 WSN

中同时运行多个路由协议，那么做出路由协议选择决策就是一种操作。

13.3 WSN网络编程

目前，大多数 WSN 应用是采用复杂、低级程序实现的，这种程序详细指定了单个传感器节点的行为。当前 WSN 编程最流行的平台是 TinyOS。WSN 的嵌入式特性要求在网络中传播新的程序代码。

需要高级、全面的编程模型，可以利用这种模型按照系统范围内行为详细说明应用，然后自顶向下编译成每个装置的程序。WSN 编程面临的挑战是协调网络中的感知、联合处理以及数据流，实现所要求的功能。当采用手工方式将全部应用行为翻译成每个节点的本地行为时，应用逻辑与低级服务（比如资源管理、路由、定位等）协调程序密切相关，系统级代码和应用级代码之间没有分开，导致 WSN 编程非常复杂。单个传感器节点有限的计算能力、通信能力、可用能量使得 WSN 编程很困难：将系统范围内的复杂行为分解成每个节点必须采取的本地行为。这种方法是自顶向下的方法，有可能大幅度减轻大规模复杂系统编程的负担。

目前 WSN 编程模型的重点在于 WSN 的配置、传播、累积。将传感器节点的功能单元定义为角色、任务、服务。问题的高级表示可以是查询、服务请求、合成事件、正式表示式。高级语言的表示式是一个重要方面，还需要分布式数据处理体系架构。必须通过编程将端用户语义请求正确翻译成 WSN 中的操作，这就需要语言表示式，请求和操作之间必须精确映射，需要实体论。

WSN 编程引起两个主要问题：编程抽象和编程支持。编程抽象的重点是给程序员提供传感器抽象和传感器数据抽象；编程支持提供单一化程序执行的额外运行时间机制，例如运行机制包括安全代码执行、可靠代码分发。

在 WSN 程序开发中，由于 WSN 资源很有限、传感器节点易于失效且高度分散、共享的大量信息需要联合处理，所以不能在真实 WSN 上调试程序。

13.3.1 编程抽象

编程抽象分成两类：应用级编程抽象和系统级编程抽象。应用级编程抽象以所需要的语义抽象等级定义和控制事件（比如目标的最新位置、所有故障传感器的位置）；系统级编程抽象精确说明分布式计算和通信（比如对 10 m 范围内的所有 x 进行 $f(x)$ 运算，将数据条目 d 发送给 10 个最近的节点）。这两种编程抽象模型之间的折中平衡就是表示法、效率、可重复使用性、自动操作。

现有的编程抽象大致分成以下几类：基于数据的模型、基于代理的模型、宏编程模型。

1. 基于数据的模型

基于数据的模型遵循传统的基于查询的方法，代码分成两部分：一部分是服务器一方的代码，处理查询分析、查询规划、查询优化；另一部分是传感器一方的代码，负责路由、查询累积、局部数据累积、寿命规范等。这种方法是传统查询的直接扩充，增加了针对 WSN

的查询语言的修改，将消息传输路由和数据累积综合在一起。这种模型适用于流数字数据的收集。

其中最令人感兴趣之处是实现更高效率的跨层修改。跨层法在传统分层网络协议模型中很少使用，但是有利于资源有限情形和追求简易实现。图 13-1（a）将 WSN 表示为一个分布式数据库。

2. 基于代理的模型

程序员在编写一个代理时可以读/写代理本地堆栈中的变量和节点本地变量。代理可以移植到另一个节点。代理可以读入传感器数值、激活装置、发送任意无线分组。在基于代理的模型中可以实现任何算法。代理传播（在某种意义上就是路由）是程序员做出的决策，其代码在代理自身转发逻辑中。WSN 中的激励器需要相互协作控制，共同实现同一个目标。可以采用正式语言定义感知单元和激励单元应该遵循的条件和作出的反应。采用正式语言能够简化分布式系统编程，通过正式分析和自动推理提供简易性。图 13-1（b）表示移动代理（活动传感器）模型。

3. 宏编程模型

宏编程（Macroprogramming）是指对大规模复杂 WSN 的一种编程方法。宏编程的目标是通过提供高级编程抽象和原语简化 WSN 应用设计，自动向下编译成每个传感器节点的复杂低级操作。宏编程考虑的是一个 WSN 网络的全部行为，而不考虑单个传感器节点的低级行为。端用户详细说明高级任务，隐藏节点通信协议的嵌入式系统细节。本章稍后详细介绍一个抽象任务图编程模型（ATaG）。

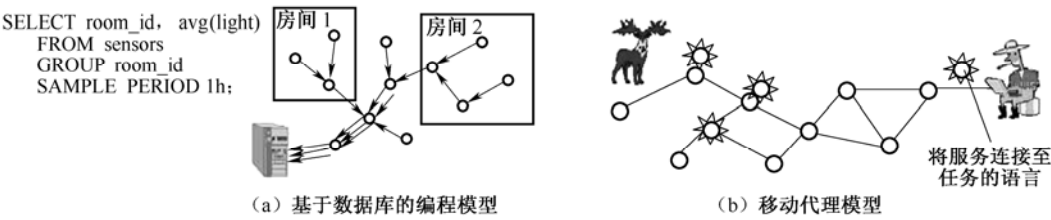


图 13-1 编程抽象

13.3.2 现有若干编程模型简介

现有宏编程可以分成两类：一类重点在于给程序员提供抽象，抽象简化说明节点本地行为的任务（在分布式计算范围内）；另一类就是使程序员能够分布式计算的全部行为。

1. Kairos

Kairos^[32]提供的网络编程模型允许程序员按照集中方式表示整个 WSN 分布式计算的全部行为。Kairos 编译时间和运行时间子系统影响一小部分编程原语，对程序员隐藏分布式代码生成和实例、远端数据访问和管理的细节，隐藏节点程序流协调。Kairos 不包含明确的节

点抽象，但是以不受网络约束的方式表示分布式计算。图 13-2 给出了 Kairos 编程体系结构。

2. Mate

Mate^[36]是一个在 TinyOS 上运行的字节代码解释程序(虚拟机),提供安全程序执行环境、运行时重复编程以及事件驱动基于堆栈的体系结构。Mate 提供虚拟机能够解释和执行的高级指令(比如发送极短消息)。每条虚拟机指令执行其自己的 TinyOS 任务。代码封装后自行分发控制识别码和版本号。Mate 维护一个固定执行事件集合、有限汇编级编程以及单个集中共享变量,提供可靠性。采用 Mate 的 WSN 重复编程过程很简单。当设计了一个新程序并进行配置时,给这个新程序分配一个新版本号,并将这个新版本号广播给 WSN。Mote 传感器接收到一个新版本的代码封装后立即安装到自己节点上,然后转发给自己的相邻节点。采用广播将新程序分发到整个 WSN 中。

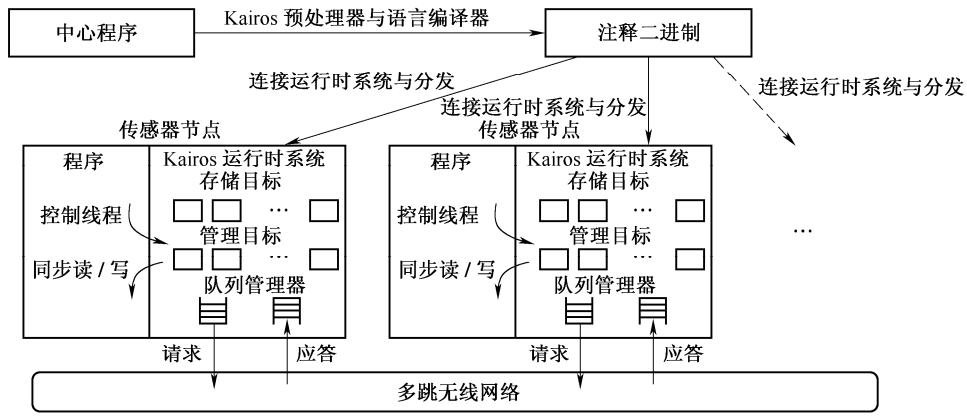


图 13-2 Kairos 编程体系结构

3. 目标状态机

目标状态机 (Object State Machine, OSM)^[33]是基于有限状态机的传感器节点的编程模型和编程语言。OSM 提供事件触发程序复杂性管理抽象,利用状态和状态转移扩充事件范例,因此行使操作成为事件和程序状态的一个功能。OSM 引入状态属性,各个动作共享信息。可以认为状态属性是某个状态的本地变量,支持自动存储器管理。可以将 OSM 说明书编译成只需要最低运行时系统支持的连续 C 代码,得到高效而紧凑的系统。OSM 借用了状态机的分层和并行组成概念以及状态机内事件广播通信概念。OSM 中的变量必定是状态分层,OSM 在进行计算时可以访问事件的数值。

4. Regiment

Regiment^[34,35]是 WSN 功能性宏编程语言,其基本概念类似于 Kairos,通过表示分布式计算的全部行为提供编程抽象。Regiment 中的必要数据模型以区域流为基础,代表收集空间分布、时间变化的节点状态。一个区域流可能代表某个区域内所有传感器节点的感知数据集合,或者该区域内感知数据的累积结果。区域提供传感器节点间空间与逻辑关系的表示方法,节

点间共享的透明数据表示法、区域内有效减少的操作。抽象区域影响资源使用与联合操作精确性之间的平衡关系，允许应用调整能耗和带宽以满足精确性目标。

Regiment 是纯功能性语言，有相当大余地通过传感器节点实现区域流操作和利用网络内的冗余度，允许用户执行通用操作[如映射（MAP）、交叠（FOLD）、过滤（FILTER）]，对一个区域内的全部数据做出一个功能映射、累积或者过滤。系统决定数据的存储地点和时间以及在网络中执行的操作。

13.4 分层编程与ATaG编程架构

非专业端用户很可能对网络化感知的系统级基础设施问题不感兴趣。只要方便且易于使用，提供的机制可以用来精确表达总的感知与响应行为，那么端用户就不用关心如何将机制翻译成节点级行为、每个节点具有什么能力、支持什么操作系统和编程语言、如何管理故障问题、使用什么路由协议、如何控制和协调各个节点对信道的访问等。

应用开发体系架构（**Application Development Framework, ADF**）有两个方面的作用：第一，**ADF** 封装和抽象系统级协议和服务，然后按照端用户易于理解的形式输出；第二，**ADF** 将其用户（程序员）定义的应用行为翻译成低级系统组件能够理解的形式，比如配置文件、功能参数、数据结构等。**ADF** 的组成是：①表示系统中联合感知、激励、计算、通信行为的编程模型（抽象语法与语义）；②用于输入描述的程序表示法（具体语法）；③实际管理节点上各种任务和节点之间交互的运行时系统模板；④分析高级应用表示法和相应配置目标系统每个节点的运行时系统模板的编译器。若设计好 **ADF**，则系统级基础设施服务（比如网络协议、通信库、中间件等）开发人员就能够将重点只放在提供 **ADF** 要求的接口上，而不用关心服务如何与系统其他组件的交互。在设计编程模型和编程语言时可以假定所有低层复杂性可以按照一个明确定义的、受到某个低层运行时系统管理和控制的接口集合来使用。

13.4.1 WSN的分层编程

WSN 高级服务和高级抽象的设计目标是使非专业人员易于访问计算基础。从非专业编程员角度来看 **WSN** 不同协议和服务提供的功能性，图 13-3 给出了这些功能性的归类。

最底层是由传感器节点（可能是不同种类的传感器节点）组成的网络。假定每个传感器节点至少支持一种操作系统和一个本地编译器。根据传感器节点的计算与存储能力，支持的低端操作系统可以是 **TinyOS**^[40]、**μC/OS-II**^[41]或者 **Contiki**^[42]，支持的高端操作系统可以是 **Linux**、**WindowsCE**。

可以将基础设施协议大致分成两个并行栈：感知栈、处理栈。感知栈涉及按照所表示的物理环境现象对有关一个或者多个感知接口上感知数据的解释问题，感知栈中的协议用于传感器标度和传感器故障检测以及按照感兴趣和对应用有意义的变量抽象物理感知接口。例如，**WSN** 个人卫生健康监视应用感兴趣的是人面临的压力程度，可以利用感知栈中的协议以及各种传感器获取的信息（如脉搏速率、血压、血氧等级等）计算应用级变量（压力）。还可以定义一组服务来研究传感器感知数据的时空相关性，以便压缩事件信息、执行应用指导下的传感器数据过滤、提供服务质量（**QoS**）保证等。

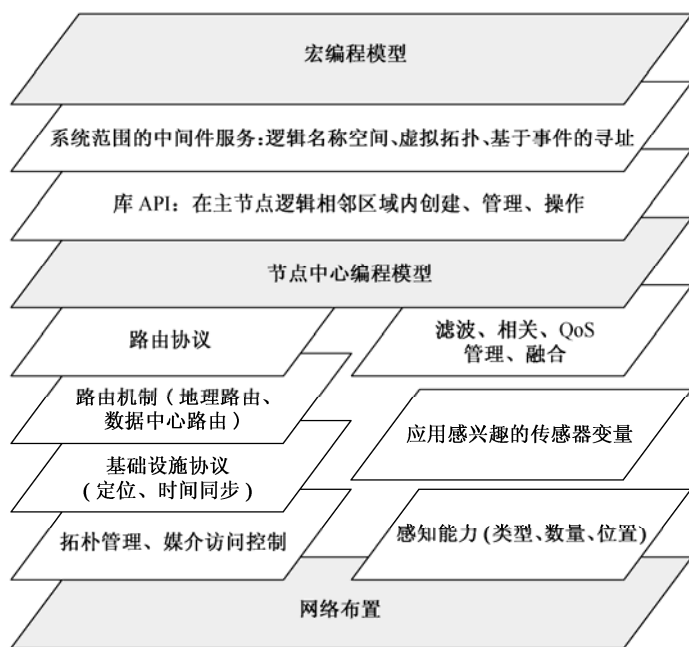


图 13-3 WSN 应用开发抽象的分层

处理栈包含提供 WSN 原始感知数据以及累积数据分布式计算与通信基础设施的协议。处理栈的低层是基本的物理层和媒介访问控制层网络协议，这些网络协议足够提供 WSN 拓扑管理的基本支持，提供相邻节点间的消息发送能力。定位协议和时间同步协议作为基本通信能力，使单个传感器节点能够根据其网络中的时空位置提供环境意识。因为只知道事件发生而不知道事件发生的物理位置和时间几乎是没有任何意义的，所以定位在 WSN 中非常重要。况且，其他大多数节点级服务都假定节点知道自己所在真实坐标系统（或者虚拟坐标系统）中的位置。位置信息支持地理路由机制，地理路由机制是 WSN 各种通信模式的基础。

感知栈和处理栈中的协议为节点中心编程提供基本的状态信息和抽象。在节点级抽象，应用开发人员必须按照每个节点的本地操作解释整个应用行为，根据每个传感器节点支持的能力、操作系统、编译器，采用诸如 nesC^[43]、galsC^[44]、C/C++或者 Java 等语言单独编写该节点的程序。程序员可以从本地感知接口读取感知数据，在本地存储器中维护应用级状态，按照节点 ID 或者位置寻址方式给其他节点发送消息，处理其他节点发送来的消息。节点中心编程可以手工进行跨层优化，完成高效实现；这种方法要求专门技能和努力，但是对于开发复杂的大规模 WSN 应用行为是不够的。

节点中心编程通用的一个概念是根据距离、转发跳数或者其他属性定义的逻辑相邻区域概念。依据逻辑相邻区域的常用操作包括：采集所有相邻节点的数据，将数据分发给所有相邻节点，对存储在相邻节点中的特定数据进行某种计算变换等。相邻区域创建与维护的有效性、普遍存在性推动了节点级库的设计和开发^[45, 46]，这些库处理控制与协调的低级细节，并且给程序员提供相邻区域 API。

中间件服务提供抽象的下一层。按照中间件对这些协议分类如下：提供系统范围抽象的协议、提供高级服务抽象的协议、提供现象中心抽象的协议。中间件服务能够创建虚拟拓扑（比如网状拓扑、树等），按照动态逻辑关系（比如主节点-成员节点、父节点-子节点等）使

程序处理其他节点，支持状态中心编程模型^[47]等。中间件协议本身通常使用节点中心编程模型来实现，并且可以使用通信库作为其实现的一个组成部分（但是不要求这样做）。

在抽象层上面，重点是详细说明累积、自动编译成单个节点的节点中心程序的 WSN 宏编程模型和语言^[32, 34]。宏编程语言一般受到低层运行时系统的支持。运行时系统管理节点级的控制与协调，其结构取决于宏编程模型的结构和语义。最终定义一个更高级的抽象，各个应用采用宏程序或者节点中心程序描述，并且按照服务来建模。与网络化传感器系统交互的端用户很可能是通过完全声明的特定域接口，根据所需的服务集将该接口编译成宏程序（或者节点中心程序）。

13.4.2 抽象任务图编程架构（ATaG）

运用 WSN 能够进行嵌入式、密集的物理环境监视。WSN 编程面临的挑战是正确且高效地协调感知、联合处理、数据流在网络中的传输，从而实现所要求的功能，满足性能要求，网络寿命最大化。大量自治传感器节点的管理需求也对 WSN 编程提出了挑战。当前的 WSN 编程语言和编程方法要求端用户手工将整个应用行为转换为每个节点的本地行为，人工转换既费时间又易于出错（特别是对于复杂应用）。应用级逻辑密切关联协调低层服务（比如资源管理、路由、定位等）的程序代码。系统级代码与应用级代码没有分隔开，导致系统行为实现现代码非常复杂。

WSN 宏编程不是节点中心编程，而是详细说明累积行为，然后由编译体系架构将累积行为自动转换成节点级规范。端用户是领域专家，而不是计算机科学家，主要对物理现象的监视和控制感兴趣，由此推动了传感器宏编程的开发。在大多数情况下，提供所需功能的网内计算和通信的详情不是端用户的主要兴趣。

抽象任务图（Abstract Task Graph, ATaG）是宏编程模型，构建数据驱动计算核心概念，并进行扩充，用于分布式感知与响应应用。将系统中信息处理功能类型模拟为一个抽象任务集合，具有良好的输入/输出接口。用户提供的有关每个抽象任务代码实现系统中的真正处理。因为抽象任务的数量和布置以及控制与协调机制都是在编译时或者运行时依据目标部署的特点而决定的，所以一个 ATaG 程序就是“抽象”。

采用 ATaG 能够进行独立于体系结构的网络化感知应用开发。体系结构独立性就是详细说明一个通用、参数化网络体系结构的行为的能力。可以自动综合不同网络部署的相同应用，相同应用能够适应节点失效或者系统增加新的节点。能够在决定节点和网络的最终配置之前继续开发应用。

ATaG 编程法的重点在于较复杂信息流模式的简单说明。ATaG 的第二个目标是定义自动分析用户提供的 ATaG 程序、生成特定布置的分布式软件系统的过程，分布式软件系统包括用户提供的用户级代码、适合用户化的运行系统（负责各个应用级任务之间的协调）。

1. ATaG 关键概念

ATaG 以两个关键概念为基础：数据驱动程序流和混合强制声明说明书。

在数据驱动计算中，任务是按照其输入输出接口定义的被动目标。任务之间不相互交互。程序员可用的基本原语是 `getData()` 和 `putData()`，前者表示使用数据池中的数据条目，后者用于生成数据池中的数据条目。当一个任务的操作数有效时，就自动安排该任务的执行，这种

安排由低层运行时系统来完成，低层运行时系统负者管理数据池。这种数据驱动计算很有吸引力，理由如下：①任务可以按照任何所需抽象等级使用数据条目，而不用考虑数据条目是如何生成的；②因为不存在任务与任务之间的直接耦合，所以程序可扩展性强、可重复使用性高；③从实现来看，数据驱动编程可以得到事件驱动运行时系统的自然支撑，导致高资源利用率。一个“事件”就是取出数据池中的一个数据条目，或者生成一个数据池数据条目。

混合强制声明说明书便于明确分开功能性、非功能性（比如任务布置、协调）。对于WSN，这种分离特别关键，因为通过解释不同的声明部分就能够将相同程序综合到各种布置中而不需要修改该程序。为混合强制声明设计可视化编程接口，因此程序员不需要掌握新语法。通过这种分离还可以支持网络意识和网络透明度^[48]。ATaG 将功能性、非功能性分开是为了设计可用于网络化感知应用的合适机制。

2. ATaG语法

ATaG 采用网络意识方法捕获整个应用行为。WSN 与物理环境的密切关联以及数据网内处理依赖处理数据的时空位置强制性需要网络意识。另一方面，可携带性、体系结构独立性要求不受网络约束的模型和表示法。ATaG 并不隐藏并行性。编译器将简明扼要的、与体系结构无关的、但是又不是明确并行的说明书翻译成节点级控制与协调行为。

一个 ATaG 程序就是一个抽象声明集合。一个抽象声明可以是如下三种类型之一：抽象任务、抽象数据、抽象信道。每个抽象声明由一个注释集合组成。每个注释是一个 2 维数组：第一个元素是注释的类型，第二个元素是数值。

(1) 抽象任务

每个抽象任务声明代表一种处理类型，在应用中可能发生这种处理。与该抽象任务声明有关的注释按照特定网络描述决定系统在给定时间存在这种抽象任务实例的数量。程序员给每个抽象任务添加一个唯一的名称。跟每个抽象任务声明有关的是一个采用目标平台支持的传统编程语言编写的可执行规范。表 13-1 描述了抽象任务声明及其注释（ATaG 当前版本）。

类型：实例	
值[：参数]	描 述
one-on-node-ID:id	在节点 ID 上创建一个任务实例
one-anywhere	在网络中任意节点上创建一个任务实例
nodes-per-instance:[/]n	对网络中每 n 个节点创建一个任务实例。当 n 由 “/” 决定时，严格创建 n 个任务实例，将网络节点总数分成 n 个非重叠域，每个非重叠域拥有一个任务实例
area-per-instance:[/]area	与 nodes-per-instance 相同。参数表示布置的区域而不是节点数；按照布置的区域而不是按照节点数表示非重叠域
Spatial-extent: x_1,y_1,x_2,y_2,\dots	对由坐标 (x_1,y_1) 、 (x_2,y_2) 、 \dots 、 (x_1,y_1) 确定的多边形中的每个节点创建一个任务实例
类型：触发规则	
值[：参数]	描 述
periodic:p	安排周期性任务（周期 p 秒）的执行时间
any-data	当至少一个输入数据条目可用时，安排执行时间
all-data	只有当所有输入数据条目可用时，才安排执行时间

(2) 抽象数据

每个抽象数据声明代表一种应用特定的数据目标类型，可以在抽象任务之间交换这种数据目标。在 ATaG 中，抽象数据声明不存在语义。与该抽象数据声明有关的注释，按照特定布置、依据抽象任务生成以及该数据目标提取的实例化和启动规则决定系统在给定时间存在某特种类型数据目标实例的数量。

程序员给每个抽象数据声明添加一个唯一的名称。类似于跟抽象任务声明有关的可执行代码，与该抽象数据声明有关的是应用特定的有效载荷。有效载荷通常由一个目标平台支持的传统编程语言编写的变量集合组成。目前，抽象数据条目不存在注释。

(3) 抽象信道

抽象信道使一个抽象任务声明关联一个抽象数据声明，不是表示数据目标刚刚产生和（或者）被某个给定抽象任务处理，而是表示该任务某个特定实例感兴趣的那些数据条目类型实例。表 13-2 描述了与抽象信道有关的注释。抽象信道对于网络中信息流公共模式的简明、灵活、不受体系结构约束的说明书非常关键。例如，使用诸如“1-hop（一个转发跳）”、“local（本地）”、“all-nodes（所有节点）”之类的简单注释就能够在输入/输出信道上表达空间分发与收集模式。

表 13-2 抽象信道：注释

类型：初始化	
值	描 述
push	在生成有关抽象数据条目的每个实例的节点上，其运行时系统负责将该抽象数据条目实例发送给驻留有使用者任务合适实例的节点
pull	在驻留有使用者任务实例的节点上，其运行时系统负责从生成节点申请有关抽象数据条目的所需实例
类型：兴趣	
值[：参数]	描 述
[]local	应用于本抽象任务实例的本地数据池的信道。不允许不合格节点访问该本地数据池，但是可以协助其他合格节点
neighborhood-hops:n	适用于驻留有本抽象任务实例的节点的 n 跳相邻区域内的所有节点的信道
neighborhood-distance:d	适用于驻留有本抽象任务实例的节点的距离 d 范围内的所有节点的信道
all-nodes	适用于系统中所有节点的信道
domain	适用于本抽象任务实例拥有的所有节点的信道，与表 13-1 中的 nodes-per-instance 或者 area-per-instance 一起使用
parent	适用于驻留有本抽象任务实例的节点的父节点的信道。运行时系统影响网络虚拟树拓扑
children	适用于驻留有本抽象任务实例的节点的所有子节点的信道。运行时系统影响网络虚拟树拓扑

3. ATaG语义

程序员可用的两条基本原语 getData()和 putData()分别用于数据条目的使用和生成。运行时系统管理数据池，在使用者和生成者之间移动数据。下面简单概述 ATaG 的语义。

假如是周期性（Periodic）抽象任务，则在周期性定时器定时结束之时安排执行该抽象任务，而不管其输入数据条目的状态。每当开始执行一个抽象任务时，就将其定时器清零；而当定时器时间等于该抽象任务的周期时间时，就说定时器定时结束。假如是 any-data 抽象任

务，则只要其任意输入数据条目的一个新实例可用，就安排执行该抽象任务。其他有效触发规则就是 `periodic ∨ any-data`、`periodic ∨ all-data`。

每个正常运行的抽象任务对其每个输入数据条目必须严格调用一次 `getData()`，对其每个输出数据条目可能最多调用一次 `putData()`。从抽象任务来看，`getData()`是破坏性读取。一旦一个特定数据条目实例被某个抽象任务所读取，则立即认为从数据池中删除了该数据条目实例，直到该抽象任务涉及该数据条目实例时为止。

抽象任务执行是全力以赴的。每个应用级任务在另一个应用级任务可以开始执行之前执行完毕。依附于某个特定数据条目的抽象任务集合中的所有任务全部在其他抽象任务（依赖其任务集合中任务的输出数据条目）执行之前执行完毕。只要生成一个数据条目实例导致其有关的一个或者多个抽象任务准备就绪，那么这些抽象任务在调用 `getData()`时就会在该输入数据条目上消耗相同实例。这就意味着在已安排执行的每个附属抽象任务执行完毕之前，触发这些抽象任务的那个特定数据条目实例不会被改写，也不会从数据池中删除。其含义就是：假如生成该抽象数据的一个实例不能被改写，那么就不能保证 `putData()`成功。例如，假如抽象数据条目的生成速率大于消耗速率，那么就很可能出现这种情况。应用开发人员负责检查 `putData()`成功还是失败。

4. ATaG程序的编译

在一个特定网络上编译一个 ATaG 程序意味着按照该特定布置翻译注释以及为目标网络的每个节点生成一个配置。一个节点的配置包含如下内容：

- 分配给节点的任务集合；
- 每个任务的触发规则；
- 预计增加到该节点数据池中，或者本地驻留的某个任务生成的，或者其他节点发送的数据条目集合；
- 在该节点上生成的每个数据条目的目的节点集合。这个目的节点集合是一张节点 ID 列表，并且在以下两个条件下可以事先知道：一是假如是固定网络拓扑；二是对于动态拓扑，假如是未翻译的注释本身，但是在运行时系统中将会按需翻译该注释。

5. 数据驱动ATaG运行时系统（DART）

应用开发体系架构开发人员的任务是：①理解应用目标类的必要特点，以便定义合适的编程抽象；②理解低层网络体系结构和协议，以便为目标布置正确合成高效分布式程序。即使开发人员使用宏编程模型和语言，最终仍然必须由编译器将应用行为翻译成节点级程序集，每个节点一个程序。宏编程并不排除这个要求，只是将职责从程序员转移到编译器而已。因此降低了应用开发成本，并且在 ATaG 下应用级代码可携带和可重复使用。

对于 ATaG 之类的宏编程，低层运行时系统的设计非常关键，原因有两点：第一，运行时系统能够简化编译和代码生成过程。数据驱动 ATaG 运行时系统（Data-driven ATaG RunTime, DART）将不受应用约束的控制与协调机制、具体应用配置信息（表示特定节点在整个系统中的角色）明确分离开来。具体应用配置是局部的，在整个软件体系结构中只占一小部分，从而能够不加修改地将运行时系统的其他组件综合成一个该节点最终可执行的程序。第二，或许与第一个原因同等重要，运行时系统允许进行即插即用综合各种协议和服务。

图 13-4 (a) 是 DART 的高级模块化结构图。DART 采用 ATaG 模型对宏编程提供系统级支持,采用模块化结构,该结构能够容纳图 13-3 中各层的协议和服务,同时易于实现不同功能即插即用集成。在 DART 中,将整个功能划分成一个模块集合;每个模块提供一个明确定义的、与 DART 系统中其他模块的接口,拥有提供该功能所要求的全部数据和协议。这种模块化的 DART 结构具有许多优点。第一,减少了各个模块之间的交互和依赖性,从而大幅度地简化了应用设计。第二,一个模块的实现对于其他模块是隐藏的,这就意味着可以采用一个完全不同的协议集来提供相同的功能,但是又不会影响 DART 的其余模块。因此,可以根据各种布置方案以及硬件和网络的特点,选择合适的协议,裁减 DART 运行时系统,不必全部重新设计。第三,实质上只需要替换少量模块、剩余模块保持不变,可以使用相同的运行时系统软件进行功能仿真和实际布置。下面描述 DART 的每个组成部分。

(1) DART 的组成

从图 13-4 (a) 中看到, DART 由网络栈、网络体系结构、ATaG 管理器、数据池、分发程序、用户任务模块组成。

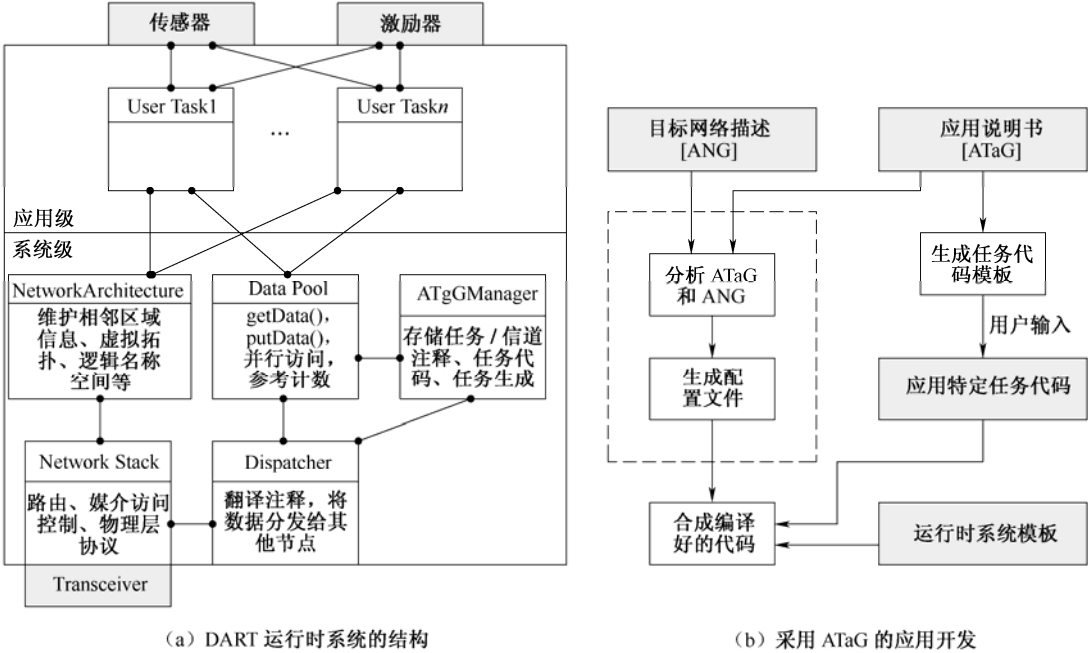


图 13-4 运行时系统、应用开发

① 网络栈 (NetworkStack)。网络栈模块负责管理传感器节点的网络接口。NetworkStack 模块提供异步的 send()和 receive()接口,实现主动消息^[49],即抽象任务在特定消息类型中记录其兴趣,这些消息到达接收方后,接收方再对其作相应处理。NetworkStack 模块有效表示 WSN 的路由协议、媒介访问控制协议、物理层协议。当对整个系统进行功能仿真时,可以在单个主机上运用套接字进行消息的发送和接收,而其余模块保持不变。

在 NetworkStack 模块中可以进行许多优化。例如,MAC 层可以采用 S-MAC 协议。S-MAC 协议安排每个节点收发信机的休眠/苏醒,达到降低总能耗的目的。一个节点在苏醒期间发送其在休眠期间缓存的所有输出数据。因为 send()是异步的,所以 NetworkStack 模块能够在收发信机休眠期间自由地缓存、合并输出数据条目,然后在收发信机苏醒工作时按照

批方式发送这些缓存数据条目。甚至可以用一个完全不同的 MAC 协议替代 S-MAC 协议，实现性能指标优化，同时又不影响 DART 的剩余模块。

② 网络体系结构 (NetworkArchitecture)。网络体系结构模块维护有关网络拓扑的信息，运行拓扑构建与维护协议，给其他模块提供逻辑相邻区域抽象，负责翻译信道注释（比如将“5 m 范围内的所有节点”翻译成一张节点 ID 列表）。对于用户级任务，逻辑相邻区域信息能够提供背景意识。对于被映射到节点上的每个任务，编译器合并为每个输入信道指定的相邻区域，将合并后的相邻区域交付给 NetworkArchitecture 模块。直观上，比如说，假如一个任务需要 4 跳范围内所有节点的一个特定数据条目，那么所需要的信息正好等于真正存在的节点数量、节点 ID、节点位置之和。

③ ATaG 管理器 (ATaGManager)。ATaG 管理器模块维持整个 ATaG 表示法，包括任务代码、任务注释、信道注释、任务与数据条目之间的 I/O 关系。当给数据池增加新的数据条目时，ATaGManager 模块根据任务触发规则以及可以输入的其他可用数据条目，负责确定准备运行哪些已分配的任务。信道注释存储作为 DART 系统的一个组成部分非常重要，这是因为能够动态、按需排除注释的歧异。因此 DART 能够适应网络拓扑的动态变化，同时仍然保留 ATaG 程序的高级目的。

④ 数据池 (DataPool)。数据池模块负责维持本地数据池和实现 getData()和 putData()函数调用。数据池管理涉及多个用户级任务或者系统级任务的并行访问，要求维持一个数据条目的每个实例的参考计数，才能确定某个特定实例是活动（即继续等到被一个或者多个已安排执行的任务用完为止）还是非活动（即当本地某个任务生成相同数据条目类型的一个新实例时、或者当 NetworkStack 模块接收到其他节点相同数据条目类型的一个新实例时，可以改写该数据条目实例）。函数 getData()将所申请数据条目的一个复制返回给调用函数，将有关条目的参数计数值减一。函数 putData()给数据池增加一个特定抽象数据条目的一个实例；但是当现有数据条目实例是活动的时候，函数 putData()不会改变数据池而直接返回。当给数据池增加一个新数据条目时，ATaGManager 模块会得到通知，从而有可能安排执行一个或者多个任务；Dispatcher 模块也会得到通知，此时该数据条目会被分发给其他节点。

⑤ 分发程序 (Dispatcher)。分发程序模块负责将在本节点上生成的数据条目分发给网络中的其他区域。Dispatcher 模块支持 notify()函数。只要本地生成一个新数据条目，DataPool 模块就调用 notify()函数，然后 Dispatcher 模块联系 ATaGManager 模块，确定有关该数据条目输出信道的注释。必须根据网络拓扑当前状态（即 NetworkArchitecture 模块的活动范围）将该输出信道注释翻译成一张节点 ID 列表。当进行输出信道注释翻译时，Dispatcher 模块将该数据条目（已具有目的节点 ID 列表）交付给 NetworkStack 模块。

⑥ 用户任务 (UserTask)。用户任务模块表示一个应用级任务，即 ATaG 模型中“abstraction task (抽象任务)”的一个实例。对于分配给节点的每个抽象任务存在一个 UserTask 模块实例。程序员（以及因此 UserTask 模块）访问 DataPool 模块的 getData()和 putData()接口以及 NetworkArchitecture 模块（负责将逻辑区域翻译成一张节点 ID 或者位置列表）的接口、本节点的传感器接口和激励器接口。

(2) 流程控制

流程控制分成两个部分：第一个部分是在节点初始化期间发生的活动集，第二部分是在该节点上执行应用过程中触发的活动。

DART 的每个模块均实现一个 start()函数，用于完成该模块所要求的基本初始化。初始

化可能涉及存储器分配、变量初始化、生成各种协议和服务的线程等。当一个节点加电开机时，首先运行 Startup 模块，然后依次调用其他模块的 start()函数。首先启动 DataPool 模块，主要包括对应 ATaG 中的不同数据条目为数据池中的每个数据条目分配存储器，然后将其初始化为合适的参考数，表示数据条目空。接着启动 NetworkStack 模块，生成接收输入连接关系的接收节点线程、处理输出消息的发送节点线程。若是需要，那么在 NetworkStack 模块的 start()函数将控制返回给 Startup 模块之前，还要进行 MAC 协议、路由协议、定位协议、时间同步协议所要求的初始化，然后启动 NetworkArchitecture 模块。因为已经完成 NetworkStack 模块的初始化，可用发送/接收能力，所以 NetworkArchitecture 模块能够生成相邻节点寻找、虚拟拓扑构建、中间件服务等所需要的协议。当完成累积某个最小化节点状态（比如有关相邻区域的所有可用信息）时，则认为已经完成 NetworkArchitecture 模块的启动，最后启动 ATaGManager 模块。ATaGManager 模块越过分发给该节点的用户级任务列表，生成程序员标记为“在初始化期间运行”的所有任务，这些任务通常周期性产生传感器感知数据集、传感器感知数据又驱动其余的网内处理。

在正常应用执行过程中可能发生三个主要事件：①某个用户任务调用 getData()函数；②某个用户任务调用 putData()函数；③接收节点线程接收到另一个节点的一个数据条目时调用 putData()函数。当调用 getData()函数时，DataPool 模块只是将有关的数据条目的参考数值减一。当一个本地任务调用 putData()函数时，DataPool 模块首先检查相应的数据条目是否为非活动条目，然后再决定将新生成的数据实例增加到数据池中。若在数据池中成功增加一个数据条目实例，则 DataPool 模块通知 ATaGManager 模块生成该数据。ATaGManager 模块确定依赖该数据条目的任务列表，检查其触发规则，安排执行合格的任务。然后 DataPool 模块通知 Dispatcher 模块，最终将控制返回给用户任务。Dispatcher 模块与 ATaGManager 模块、NetworkArchitecture 模块、NetworkStack 模块交互，将该数据条目发送给 ATaG 程序指定的其他节点。对于第③个事件，除了不包括 Dispatcher 模块，其他处理与本地任务调用相同。

(3) DART 讨论

DART 及其流程控制可以在满足以下条件的任何操作系统内核上实现：①支持多线程执行；②抢先式的、基于优先级的调度程序；③相互排斥的旗语、消息查询、临界区域并行访问处理机制、不同线程间相互交互的协调机制。

DART 反映了图 13-3 所示的 WSN 分层编程的分类思想，在系统中不同模块间明确划分功能性，只要接口保持相同就可以完全改变每个模块的实现而不影响其他模块。可以按照 DART 体系结构模板设计图 13-3 中各层的协议和服务。例如，新路由协议设计人员不必操心与其他中间件服务、路由协议或者应用级的交互。可以将新协议插入到 NetworkStack 模块中，NetworkStack 模块根据 ATaG 程序中指定的有关性能注释选用这个新协议来发送在本节点上产生的一部分数据条目。DART 的性能很可能比不上手工优化的运行时系统，后者将不同功能综合成一个紧凑的、不易更改的单一结构，并且进行了许多跨层优化。但是，一方面是可可用性和灵活性，另一方面是性能手工优化，这两者之间的综合平衡对于复杂系统的自动化设计方法是很常见的。

13.4.3 采用ATaG的应用开发方法

图 13-4(b)描述采用 ATaG 的应用开发方法。应用开发人员采用有注释网络图(Annotated

Network Graph, ANG) 形式、按照图形方式输入 ATaG 程序的声明部分以及目标部署的描述。ANG 包含节点数量、每个节点的坐标、网络连通性等信息。代码生成器分析 ATaG 程序, 确定抽象任务与数据目标之间的 I/O 依赖关系。程序员对代码模板增加应用功能性。然后编译器按照 ANG 解释程序注释, 生成每个节点的可配置文件, 根据每个节点在网络中的角色在配置文件中定制该节点的行为。最后, 为每个网络节点生成准备编译的代码。

对编程与合成环境的接口就是通过一个可配置图形工具套来实现, 该工具套称为通用建模环境 (Generic Modeling Environment, GME)。对 ATaG 程序的声明部分及其注释加以可视化详细说明。实际上, 利用 ATaG 程序的可编排性, 允许用户创建 ATaG 程序库, 只需将 ATaG 程序库中的程序简单连接在一起就能够建立较大的应用。GME 存储用户按照规范格式定义的模型, 可以利用被称为模型解释程序的工具读/写模型数据库。在图 13-3 中, 模型解释程序就是非阴影组成方框。

13.4.4 一个 ATaG 应用例子

图 13-5 是一个环境监视系统的 ATaG 程序。每个传感器节点配置一个温度传感器、一个大气压传感器。环境监视应用表现两个行为: 周期性计算和记录系统中的最高大气压, 周期性监视温度。假如一个节点与其相邻节点之间的温度梯度大于门限值, 那么要求该节点确认温度异常: 测量较大区域, 然后进行告警。确认有助于避免传感器故障引起的伪告警。

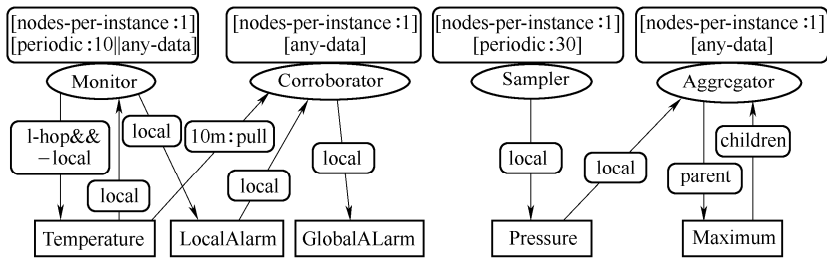


图 13-5 一个环境监视的 ATaG 程序

ATaG 程序员首先按照节点级交互模式模拟每种行为。温度监视要求相邻节点与相邻节点之间交换感知温度, 以便计算温度梯度; 温度监视需求还要求进行多点到一点的传输, 以便证实异常温度。可以按照信息流形象化大气压监视和记录, 按照每个节点虚拟树拓扑进行递增式累积, 虚拟树是高效数据累积的常用模式。

接下来是识别系统中的处理类型和数据类型, 这在 ATaG 中分别称为抽象任务和抽象数据。求平均大气压的抽象数据是: Pressure 代表大气压传感器的感知数据, Maximum 代表 (局部) 最高大气压。温度监视的抽象数据是: Temperature 代表温度传感器的感知数据, LocalAlarm 和 GlobalAlarm 分别代表门限条件和在较大区域的证实条件。大气压监视的抽象任务是: Sampler 代表节点周期性记录大气压数值, Aggregator 代表跟踪虚拟树上子节点最高大气压感知数据。温度监视的抽象任务是: Monitor 代表计算本地温度梯度, Corroborator 代表分析较大相邻区域的感知数据。程序员提供每种抽象任务和每种抽象数据的代码。这就是这个 ATaG 程序的必需组成部分, 也就是程序员编写的唯一代码。

图 13-5 给出了各个抽象任务之间的输入/输出接口。Monitor 生成代表其本地感知数据的

Temperature 实例，删除其相邻节点生成的 Temperature 实例。因为这是数据实例之间的区别而不是数据类型之间的区别，所以抽象任务和抽象数据之间的关系就是输入和输出的关系。

最后简要说明任务布置和信息流模式的有关注释（图 13-5 中的阴影圆角正方形）。例如，注释指出：Monitor 在系统中每个节点上生成实例，周期性运行，并且在新的 Temperature 实例有效时也执行。将 Monitor 生成的 Temperature 实例发送给所有一跳相邻节点，但是不添加到本地数据池中。在 LocalAlarm 作用下 Monitor 触发 Corroborator，“取出”10 m 半径范围内的所有 Temperature 实例，并且有可能产生 GlobalAlarm。对于其他行为，只要本地生成一个 Pressure 实例（通过周期性 Sampler）或者接收到本节点的子节点的 Maximum 实例，就执行 Aggregator。将 Aggregator 的输出发送到虚拟树上，虚拟树由运行时系统维护。

参 考 文 献

- [1] Beckwith, R. et al. Pervasive Computing and Proactive Agriculture. Proc. PERVASIVE, 2004.
- [2] Brennan, S.M. et al. Radiation Detection with Distributed Sensor Networks. IEEE Computer, Vol.37, No.8, pp.57-59, 2004.
- [3] Marti, M. et al. Shooter Localization in Urban Terrain. IEEE Computer, Vol.37, No.8, pp.60-61, 2004.
- [4] Martinez, K. et al. Environmental Sensor Networks. IEEE Computer, Vol.37, No.8, pp.50-56, 2004.
- [5] Britton, M. et al. The SECOAS project: Development of a self organizing, wireless sensor network for environmental monitoring. Proc. workshop SANPA, 2004.
- [6] Burrell, J. et al. Vineyard Computing: Sensor Networks in Agricultural Production. IEEE Pervasive Computing, Vol.3, No.1, pp. 38-45, 2004.
- [7] Chaudhary, S. et al. Architecture of Sensor based Agricultural Information System for Effective Planning of Farm Activities. Proc. SCC, 2004.
- [8] Shum, L. et al. Distributed Algorithm Implementation and Interaction in Wireless Sensor Networks. Proc. workshop SANPA, 2004.
- [9] Szewczyk, R. et al. An analysis of a large scale habitat monitoring application. Proc. SenSys, pp.214 - 226, 2004.
- [10] Zhang, W. et al. Optimizing Tree Reconfiguration for Mobile Target Tracking in Sensor Networks. Proc. Infocom, 2004.
- [11] Demers, A. et al. The Cougar Project: a work-in-progress report. ACM SIGMOD, Vol.32, No.4, pp.53-59, 2003
- [12] He, T. et al. Energy-efficient surveillance system using wireless sensor networks. Proc. MobiSys, 2004.
- [13] Blumenthal, J. et al. Wireless Sensor Networks – New Challenges in Software Engineering. Proc. ETFA, 2003.
- [14] Blumenthal, J. et al. SeNeTs-Test and Validation Environment for Applications in Large-Scale Wireless Sensor Networks. Proc. INDIN, 2004.
- [15] Borcea, C. et al. Spatial programming using smart messages: Design and implementation.

- [16] Heizelman, W. et al. Adaptive protocols for information dissemination in wireless sensor networks. Proc. Mobicom, 1999.
- [17] Daniel, D. et al. Introduction to high level synthesis. IEEE Des. Test, vol.11, no.4, pp.44-54, 1994.
- [18] Gui, C. et al. Power conservation and quality of surveillance in target tracking sensor networks. Proc. MobiCom, 2004.
- [19] Essa, I.A. et al. Ubiquitous Sensing for Smart and Aware Environments. IEEE Personal Communications, pp.47-49, 2000.
- [20] Holmquist, L.E. et al. Building Intelligent Environments with Smart-Its. IEEE Computer Graphics and Applications, vol.24 No.1, pp.56-64, 2004.
- [21] Kang, P. et al. Smart Messages: A Distributed Computing Platform for Networks of Embedded Systems. The Computer Journal, Vol. 47, No. 4, pp.475-494, 2004.
- [20] Schramm, P. et al. A Service Gateway for Networked Sensor Systems. IEEE Pervasive Computing, Vol.3, No.1, pp.66-74, 2004.
- [23] Wanat, R. et al. Enabling Ubiquitous Sensing with RFID. IEEE Computer, pp. 84-86, 2004.
- [24] Abdelzaher, T. et al. EnviroTrack: Towards an environmental computing paradigm for distributed sensor networks. Proc. ICDCS, 2004.
- [25] Basagni, S. et al. Secure Pebblenets. Proc. MobiHoc, 2001.
- [26] Bergamo, P. et al. Collaborative Sensor Networking Towards Real-Time Acoustical Beamforming in Free-Space and Limited Reverberance. IEEE Transactions on Mobile Computing, Vol.3, No.3, pp.211-224, 2004.
- [27] Fang, Q. et al. Locating and Bypassing Routing Holes in Sensor Networks. Proc. Infocom, 2004.
- [28] Grimm, T. et al. A system architecture for pervasive computing. Proc. ACM SIGOPS European Workshop, 2000.
- [29] Estrin, D. et al. Next Century Challenges: Scalable Coordination in Sensor Networks. Proc. MobiCom, 1999.
- [30] Bulusu, N. et al. Self-Configuration Localization Systems. PhD Dissertation, UCLA, 2002.
- [31] Hoblos, G. et al. Optimal design of fault tolerant sensor networks. Proc. IEEE IEEE International Conference on Control Applications, 2000.
- [32] Gummadi, R. et al. Macro-programming Wireless Sensor Networks using Kairos. Proc. DCOSS, 2005.
- [33] Kasten, O. et al. Beyond Event Handlers: Programming Wireless Sensors with Attributed State Machines. Proc. IPSN, 2005.
- [34] Newton, R. et al. Region Streams: Functional Macroprogramming for Sensor Networks. Proc. DMSN, 2004.
- [35] Newton, R. et al. Building up to Macroprogramming: An Intermediate Language for Sensor Networks. Proc. IPSN, 2005.
- [36] Levis, P. et al. Mate: a tiny virtual machine for sensor networks. Proc. USENIX/ACM

ASPLOS X, 2002.

- [37] R"omer, K. et al. Generic role assignment for wireless sensor networks. Proc. ACM SIGOPS European Workshop, 2004.
- [38] Bakshi, A. et al. The Abstract Task Graph: A Methodology for Architecture-Independent Programming of Networked Sensor Systems. Proc. EESR, pp.19-24,2005.
- [39] Bakshi, A. et al. System-level Support for Macroprogramming of Networked Sensing Applications. Proc. Pervasive, 2005.
- [40] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for networked sensors. in 9th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, 2000.
- [41] uC/OS-II RTOS, <http://www.ucos-ii.com/>,.
- [42] A. Dunkels, B. Gronvall, and T. Voigt. Contiki - a lightweight and flexible operating system for tiny networked sensors. in 1st IEEE Workshop on Embedded Networked Sensors, 2004.
- [43] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler. The nesC language: A holistic approach to networked embedded systems. in Proceedings of Programming Language Design and Implementation(PLDI), 2003.
- [44] E. Cheong and J. Liu. galsC: A language for event-driven embedded systems. in Proc. Design, Automation and Test in Europe, 2005.
- [45] K. Whitehouse, C. Sharp, E. Brewer, and D. Culler. Hood: a neighborhood abstraction for sensor networks. in 2nd Intl. Conf. on Mobile systems, applications, and services, 2004.
- [46] M. Welsh and G. Mainland. Programming sensor networks using abstract regions. in First USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI), March 2004.
- [47] J. Liu, M. Chu, J. Liu, J. Reich, and F. Zhao. State-centric programming for sensor-actuator network systems. IEEE Pervasive Computing, 2003.
- [48] S. Haridi, P. V. Roy, P. Brand, and C. Schulte. Programming languages for distributed applications. New Generation Computing. 16(3):223–261, 1998.
- [49] T. Eicken, D. Culler, S. Goldstein, and K. Schauer. Active messages: A mechanism for integrated communication and computation. in 19th Intl. Symposium on Computer Architecture, 1992.